PL-TR-91-2276(II) AD-A244 420

2

DEVELOPMENT OF A MAGNETOSPHERIC SPECIFICATION MODEL
Volume II: Appendices A, B, C, D, E

J. W. Freeman
R. A. Wolf
R. W. Spiro
G-H. Voigt
B. A. Hausman
B. A. Bales
R. V. Hilmer
A. Nagai
R. Lambour

Rice University
Department of Space Physics & Astronomy
Houston, TX 77251

30 June 1991

Final Report Volume II
15 November 1987-31 March 1990

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PHILLIPS LABORATORY
AIR FORCE SYSTEMS COMMAND
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731-5000

This technical report has been reviewed and is approved for publication.

MICHAEL A. HEINEMANN
Contract Manager
Space Plasmas and Fields Br.
Space Physics Division

NELSON C. MAYNARD, Chief
Space Plasmas and Fields Br.
Space Physics Division

FOR THE COMMANDER

RITA C. SAGALYN
Director
Space Physics Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/TSI, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE: 30 June 1991 | 3. REPORT TYPE AND DATES COVERED: Final(15 Nov 1987-31 Mar 1990) **Vol II**

4. TITLE AND SUBTITLE

Development of a Magnetospheric Specification Model
Volume II: Appendices A, B, C, D, E

5. FUNDING NUMBERS

PE 62101F
PR 2688
TA 04
WU GB

Contract:
F19628-87-K-0001

6. AUTHOR(S)

J. W. Freeman, R.A. Wolf, R. W. Spiro, G-H. Voigt, B. A. Hausman, B. A. Bales, R. V. Hilmer, A. Nagai, R. Lambour

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Rice University
Department of Space Physics & Astronomy
Houston, TX 77251

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Phillips Laboratory
Hanscom AFB, MA 01731-5000

Contract Manager: Michael Heinemann/PHG

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

PL-TR-91-2276 (II)

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE

Rice University has developed a practical computer model that is capable of specifying electron and ion fluxes in the middle magnetosphere during geomagnetic storms. The model, called the Magnetospheric Specification Model (MSM), uses ground-based and satellite data from the Space Forecast Center-Environment Data Base to establish initial and boundary conditions and to determine input parameters for the magnetic and electric field models. These input values are updated every 15 minutes, and new output fluxes are computed for the same times. The primary function of the MSM is the specification of fluxes of 1-100 KeV electrons in the geostationary orbit region. However, it is also designed to specify a broad range of additional parameters for the global ionospheric-magnetospheric system, including fluxes of 1-50 KeV ions, auroral electron precipitation and ionospheric electric fields.

14. SUBJECT TERMS

Electron fluxes, Ion fluxes, Magnetosphere, Spacecraft charging, Radiation belts, Space radiation, Trapped radiation, Magnetic storms, Proton fluxes, Geomagnetic storms, Energetic particles

15. NUMBER OF PAGES: 414

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

CONT OF BLOCK 13:

The model is accompanied by an application program that allows specification of fluxes at an arbitrary point in the magnetosphere within the modeling region. Consistent with its primary function, the MSM has been tested against spacecraft data for 2 substantial storms and has been shown to produce a good characterization of the enhancements of 40 KeV electron fluxes in the equatorial plane. The model never failed to predict high fluxes when they were observed, although it did predict high fluxes in some cases when they were not observed and it did fail to predict flux dropouts observed by the spacecraft. The MSM is ready for adaptation for use in an operational setting where the goal is real-time and retrospective specification of hazardous charged particle fluxes associated with geomagnetic storms.

Appendix A

FUNCTIONAL DESCRIPTION

Functional Description

2.2 Objectives. The Magnetospheric Specification Model was developed as a set of algorighms to be developed into a real-time program for use in the Space Forecast Center operated by the Air Weather Service of the U.S. Air Force. It is intended to help the Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.

2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.4 Proposed Methods and Procedures.

The most sophisticated type of theoretical inner-magnetosphere/ionosphere model is exemplified by the Rice Convection Model (RCM). However, the Air Force requirement for near-real time data output and the relatively slow run-time of the RCM made it unsuitable for Air Force opertional use. With these constraints in mind we dropped the self-consistent treatment and have separate ionospheric electric field and magnetospheric magnetic field models to trace particles in the magnetosphere equatorial plane (See attached flow chart). The magnetic field model provides the mapping between the two regions with the condition that the equatorward edge of the auroral zone in the ionosphere maps to the inner edge of the plasma sheet in the magnetospheric equatorial plane.

3.1 Specific Performance Requirements.

The output of the model includes fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region and energy fluxes and characteristic energies of the electrons and ions precipitated into the ionosphere. The throughput time will vary depending on the number and energy of particles traced, as well as the length of time for which the model is to be run.

3.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

3.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database.

3.2 Functional Area System Functions. The major components of the system are:

MSMCON	Main program - sets start, ending, and time steps for the run
PPTCON	Executive routine to control flow of Magnetospheric Specification Model
EBCON	Executive routine to control calculation of electric and magnetic field models
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time = TSTART to time = TSTOP

Attached is flow chart of each of the major components.

3.3 Inputs and Outputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day). January 1 is day 1.
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day) January 1 is day 1.
IEND(2)	End Time (seconds)
IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time step increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following are the input files required, including those to be replace by calls to the environmental data base:

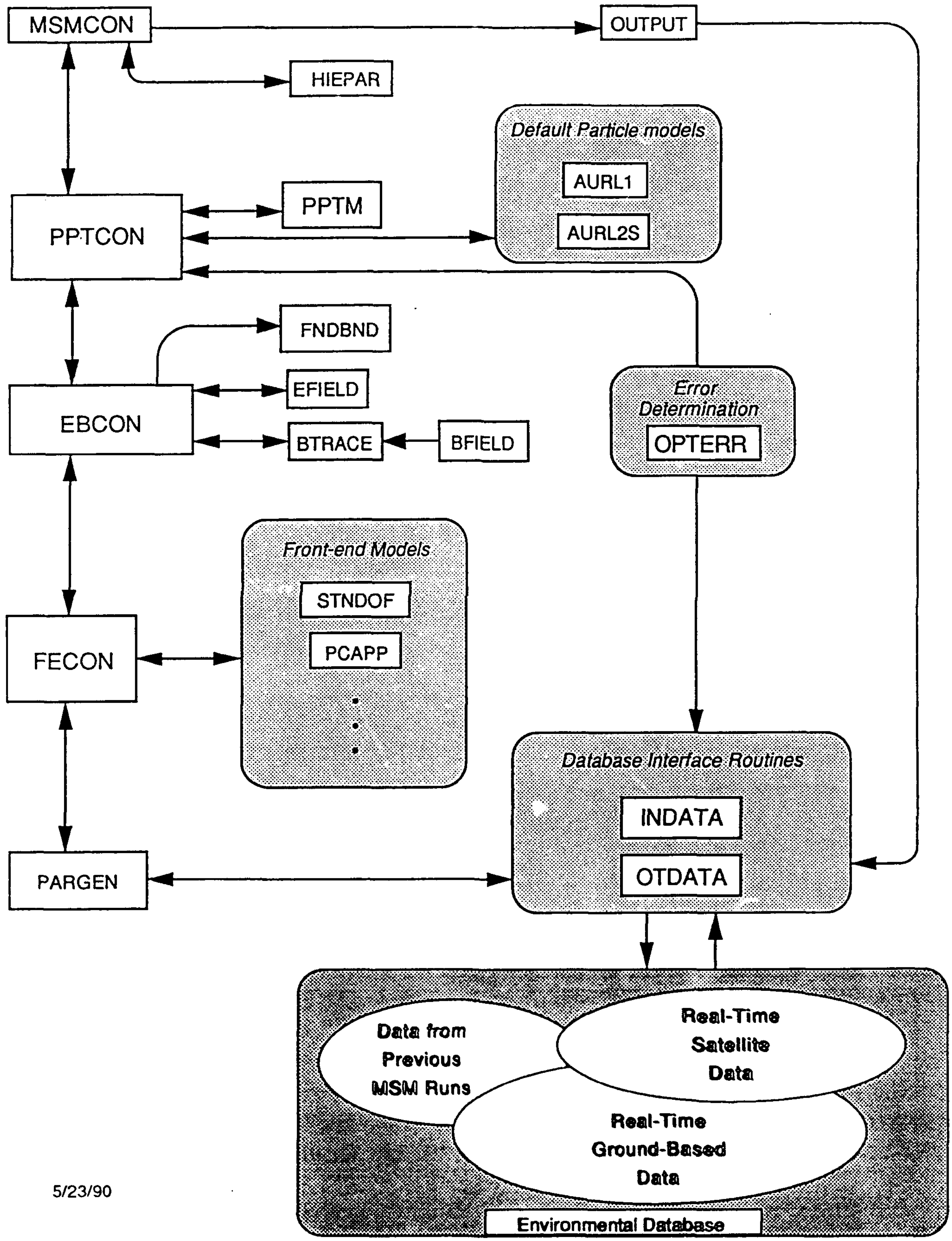
BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.

ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Following are the output files:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume $((Re/nT)^{-2/3})$.
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm^2-sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) (ergs/cm^2-sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOCT	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

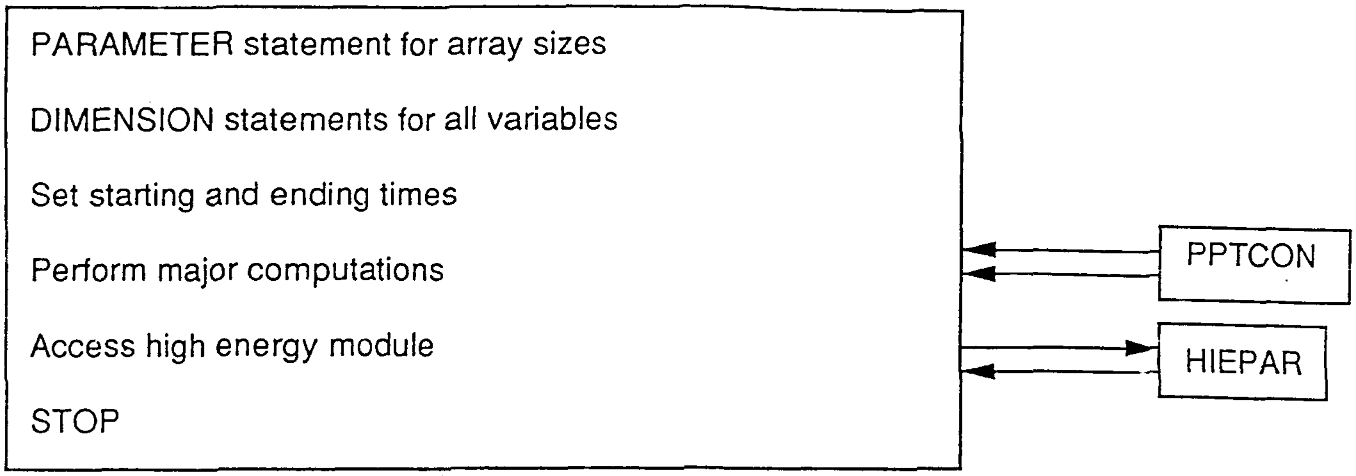
MSM Control & Data Flow Diagram



5/23/90

MSMCON

Main MSM control program



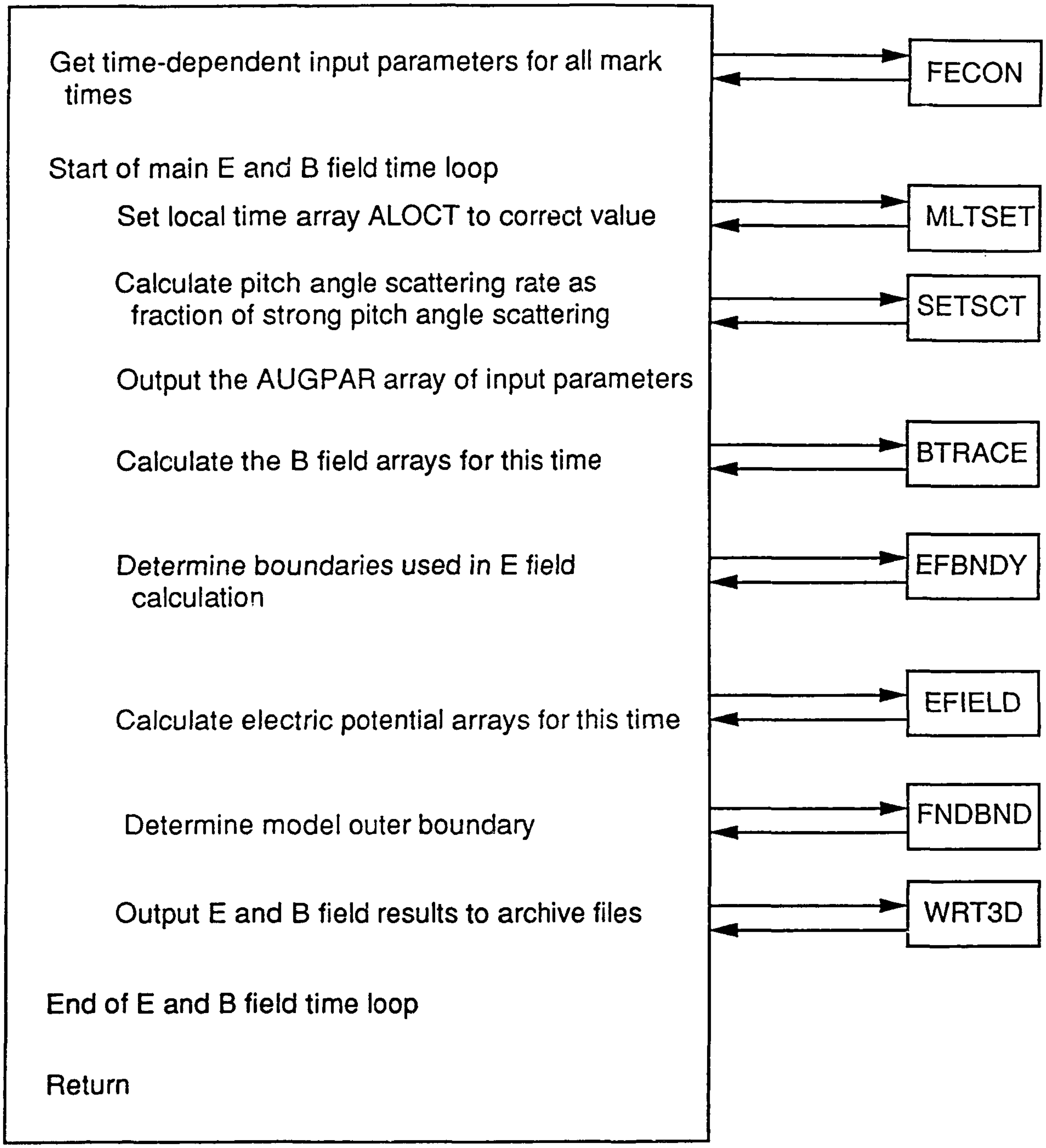
PPTCON

Subroutine that controls the major computations of the MSM. This routine calls EBCON to construct E and B field matrices at mark times and then proceeds to trace particle trajectories from each grid point for each mark time.

Set up grid	RDGRID
Initialize matrix of Kp-dependent default fluxes	
Set up energy channels to be modeled	SETALM
Initialize matrix of Kp-dependent default fluxes	FLXNIT
Compute E and B field arrays for all mark times	EBCON
Compute time-dependent plasma boundary conditions for all mark times	BNDSET
If results from previous run are available, read these results to set initial plasma distribution	RDHDR READ3D
Set initial plasma distribution	INITAL
Start main particle trace time loop	
If there is enough input data	
Do full particle traces	PPTM
Else	
Invoke default Kp-based particle distribution	FLXDFL
End if	
Calculate precipitating electron and ion fluxes from Hardy models	AURL1 AURL2S
Write results to archive files	WRT3D
Copy current values in ETA to ETABEG	
End main particle trace time loop	
Return	

EBCON

Calculate E and B field arrays at grid points for mark times



PPTM

Subroutine to trace particle trajectories from each grid point for each particle species back in time one major time step from time = TSTART to time = TSTOP. (TSTART > TSTOP)

Calculate reference flux array	SETREF
Loop over all particle species	
Loop over all grid points	
Determine normalized time BBT	TNORML
Set ETA to boundary value of eta for points outside of boundary	
Calculate locations and energies at TSTART	G3NTRP
Start adaptive Runge-Kutta time loop	
Move particle a time step and increment time	MOVER
Interpolate to find location, flux tube volume and energy at end of small RK time step	G3NTRP
Calculate loss rate this time step	
If time boundary is crossed (ie., T<TSTOP) exit Runge-Kutta time loop	TCHK
If spatial boundary is crossed exit Runge-Kutta time loop	BNDCHK
Else go to beginning of Runge-Kutta time loop	
End of Runge-Kutta time loop	
Calculate ETA at grid point from traceback and loss information. ETA is not allowed to drop below equilibrium flux value. Loss depends on threshold for strong pitch angle scattering.	
Calculate precipitating flux and energy	
End loop over grid points	
End loop over species	
Return	

SYSTEM/SUBSYSTEM SPECIFICATION

System/Subsystem Specification

2.1 System/Subsystem Description. Attached is a flow chart of the major components of the Magnetospheric Specification Model. The major components of the model are MSMCON, PPTCON, EBCON, and PPTM.

2.2 System/Subsystem Functions.

MSMCON	Main program - sets start, ending, and time steps for the run
PPTCON	Executive routine to control flow of Magnetospheric Specification Model
EBCON	Executive routine to control calculation of electric and magnetic field models
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time = TSTART to time = TSTOP

2.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

2.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database. No file interrogation takes place during the run.

4.3 System Data. The following files are necessary to restart the program:

BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.

4.3.1 Inputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day)
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day)
IEND(2)	End Time (seconds)
IINC(1)	Program time increment (year)
IINC(2)	Program time increment (day)
IINC(3)	Program time increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following is the input files required, including those to be replace by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).

SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

4.3.2 Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Equatorial magnetic field strength (nT). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces. Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)
EFLUX	Precipitating energy flux array (ergs/cm^2-sec). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons

	(2) H^+ and (3) O^+) (ergs/cm^2-sec). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
AUGPAR	Augmented data array for input values. Output volume -number of data elements × number of time steps × 4 (bytes)
COLAT	Grid colatitude array (radians). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ALOCT	Grid local time array (radians eastward from noon). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
FLUX	Flux values for all energy channels at all grid points Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

4.3.3 Database/Data Bank. The attached chart lists the parameter inputs from the operational Environmental Database and a proposed data transfer method.

4.4 Software Unit Descriptions.

4.4.1 MSMCON - see attached flow chart.

PPTCON	Main control program for model
HIEPAR	High energy electron subroutine, currently a dummy routine pending computer code from Dan Baker, GSFC

4.4.2 PPTCON - see attached flow chart.

RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities
OUTP	Utility subroutine to print out array information
FLXNIT	Subroutine to calculate Kp-based particle number flux in units of particles/cm^2/s/sr/eV as function of R, energy and Kp
DTIME	Routine which returns current system time
EBCON	Executive routine to control calculation of electric and magnetic field models
EPREAD	Dummy routine to read satellite flux data - to be replaced by call to environmental data base when program is operational
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program

BNDSET	Subroutine to compute boundary plasma distribution
RDHDR	Subroutine to read the header record of the standard MSM disk file format
INITAL	Subroutine to compute initial plasma distribution
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP
FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array - this routine is used as the MSM default model when full particle traces are not done
AURL1	Subroutine to determine default Hardy precipitating electron flux
AURL2S	Subroutine to determine default Hardy precipitating ion flux
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

4.4.3 EBCON - see attached flow chart.

FECON	Subroutine to obtain time-normalized observational values
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid points
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency
BTRACE	Subroutine to calculate magnetic field models appropriate for geophysical conditions as specified by standoff distance, equatorward edge of the auroral zone, Dst, tail collapse, and tilt angle
EFBNDY	Subroutine to calculate latitudes and widths of electric field model boundarys 2 and 3
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V)
OUTP	Utility subroutine to print out array information
FNDBND	Subroutine to compute time dependent outer boundary location
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
VMLSET	Subroutine to calculate VMLOSS array
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

4.4.4 PPTM - see attached flow chart.

SETREF	Subroutine to set upper limit reference flux
OUTP	Utility routine to print our array information
TNORML	Utility function subprogram to calculate run-normalized time
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
WKRATE	Function subprogram to evaluate weak precipitation loss rate
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction

MODFIX	Subroutine to adjust BBJ to be between 2 and JDIM-1
TCHK	Subroutine to check whether time boundary was crossed
BNDCHK	Subroutine to determine whether particle has crossed boundary of calculation
FLXVAL	Function subprogram to calculate Kp-dependent flux value a R=RR for particles of energy ENRG and Kp=FKP by interpolating FLXMAT array

PARAMETER	NAME	UNITS	I/O	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
INPUT ONLY:								
K_p	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(I)	EFLX(I)	$ELEC/CM^2\text{-}S$	I	X				X
GEO ION FLUX - CHANNEL(I)	GFLX(I)	$IONS/CM^2\text{-}S$	I	X				X
SOLAR WIND VELOCITY	SWVEL	KM/S	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	$PROT/CM^3$	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFBY	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

Data Transfer Method

1. MSM will access the environmental database input data files by calling a subroutine, INDATA.
2. The call will be of the form:
CALL INDATA(parm-name, STARTT, ENDT, NDIM, DARRY, NUMNUM)
Where:
parm-name is a valid parameter name from the MSM input parameter list.
STARTT is the starting time for this data request.
ENDT is the ending time for this data request.
NDIM is the horizontal dimension of DARRYand the maximum number of data points that may be returned.
DARRY is a two dimensional array to be retrieved from the database. The form of DARRY is shown on the next page. The size of DARRY will be NDIM by 7.
NUMNUM is the number of data values placed in DARRY by INDATA.

3. The system controller will periodically update the input data files.

4. Rice does not plan to write INDATA.

5. All parameters will be in single precision, floating point format.

6. Output data will be passed from MSM to the environmental database via a subroutine, OTDATA.

7. The call to OTDATA will be defined similarly to INDATA.

8. Rice does not plan to write OTDATA.

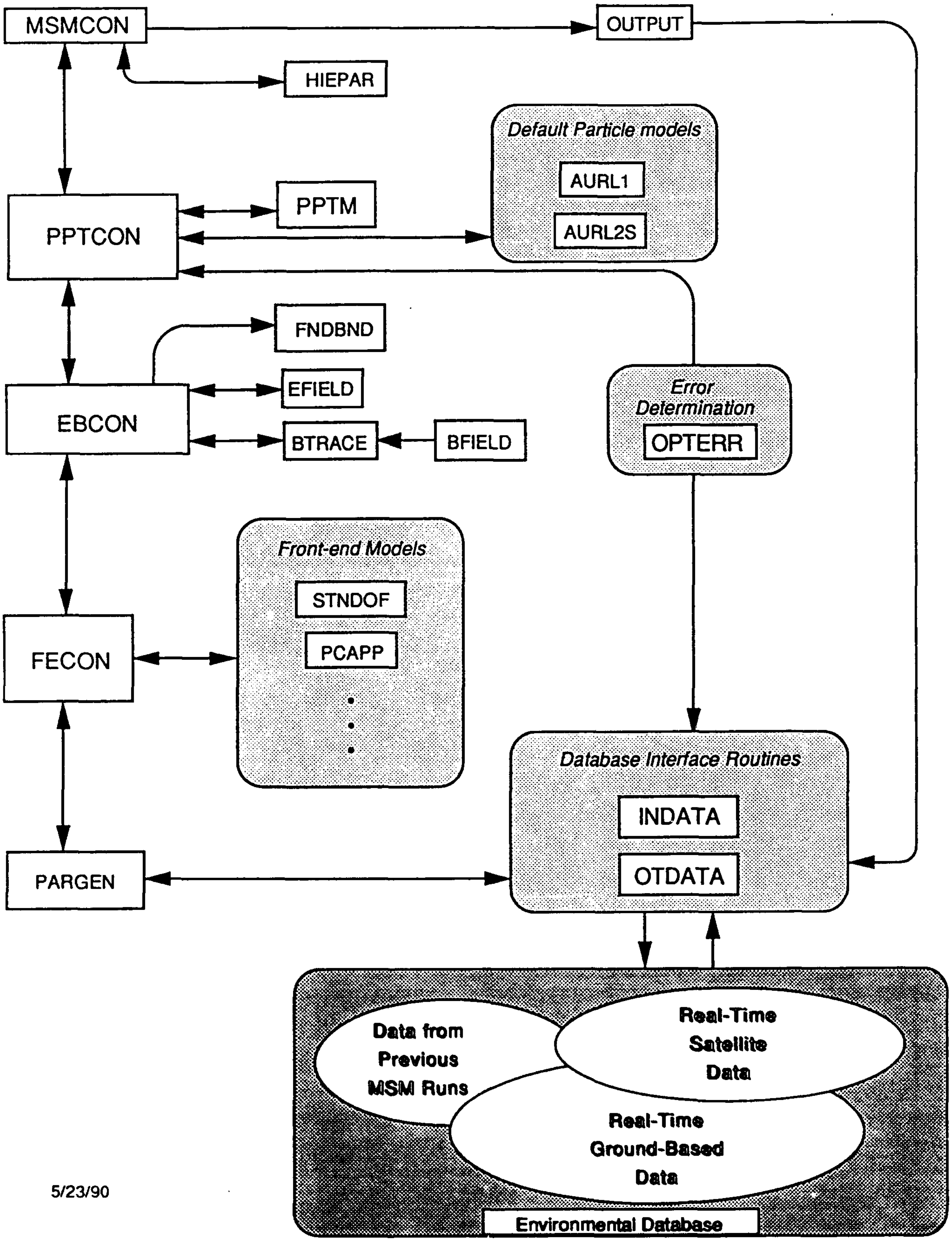
Format for DARRY

NDIM ⟶

7 ↓

Data value at t1	Data value at t2	Data value at t3	Etc.
Time tag at t1	Time tag at t2	Time tag at t3	
Latitude	Latitude	Latitude	
Longitude	Longitude	Longitude	
Altitude	Altitude	Altitude	
mag. local time	mag. local time	mag. local time	

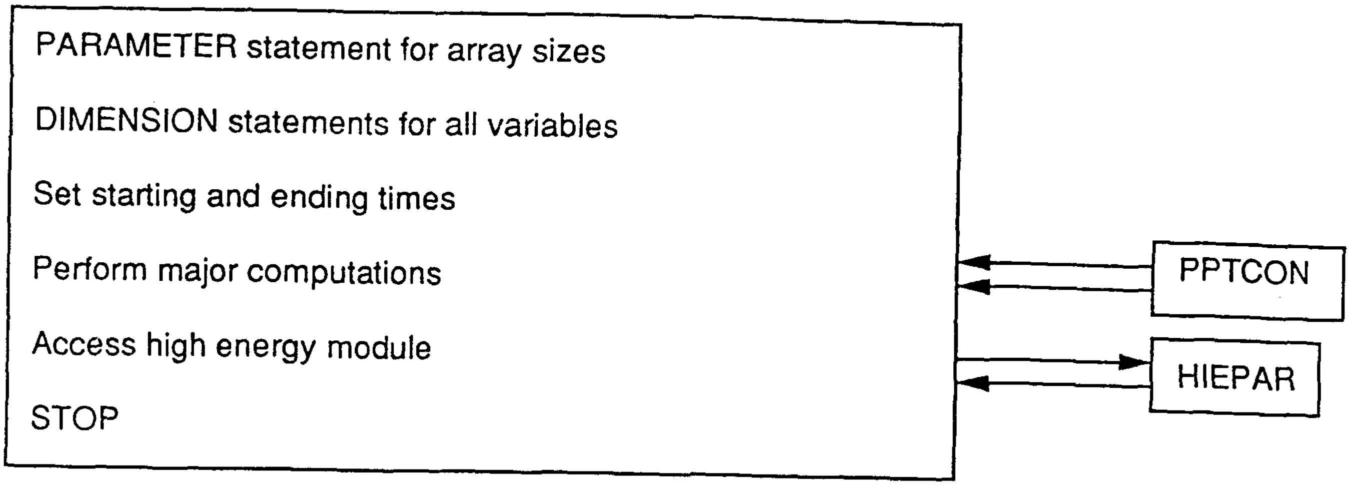
MSM Control & Data Flow Diagram



5/23/90

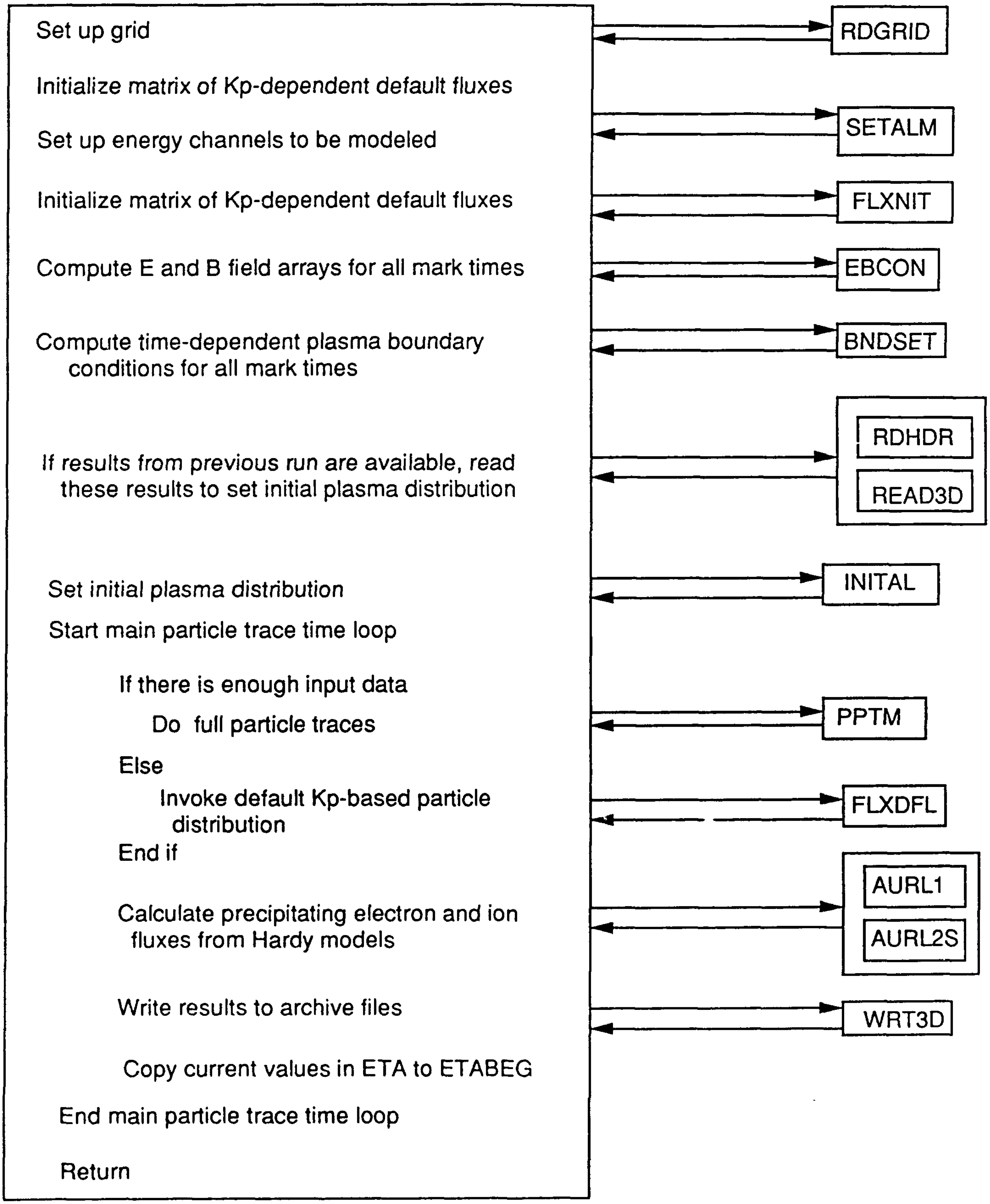
MSMCON

Main MSM control program



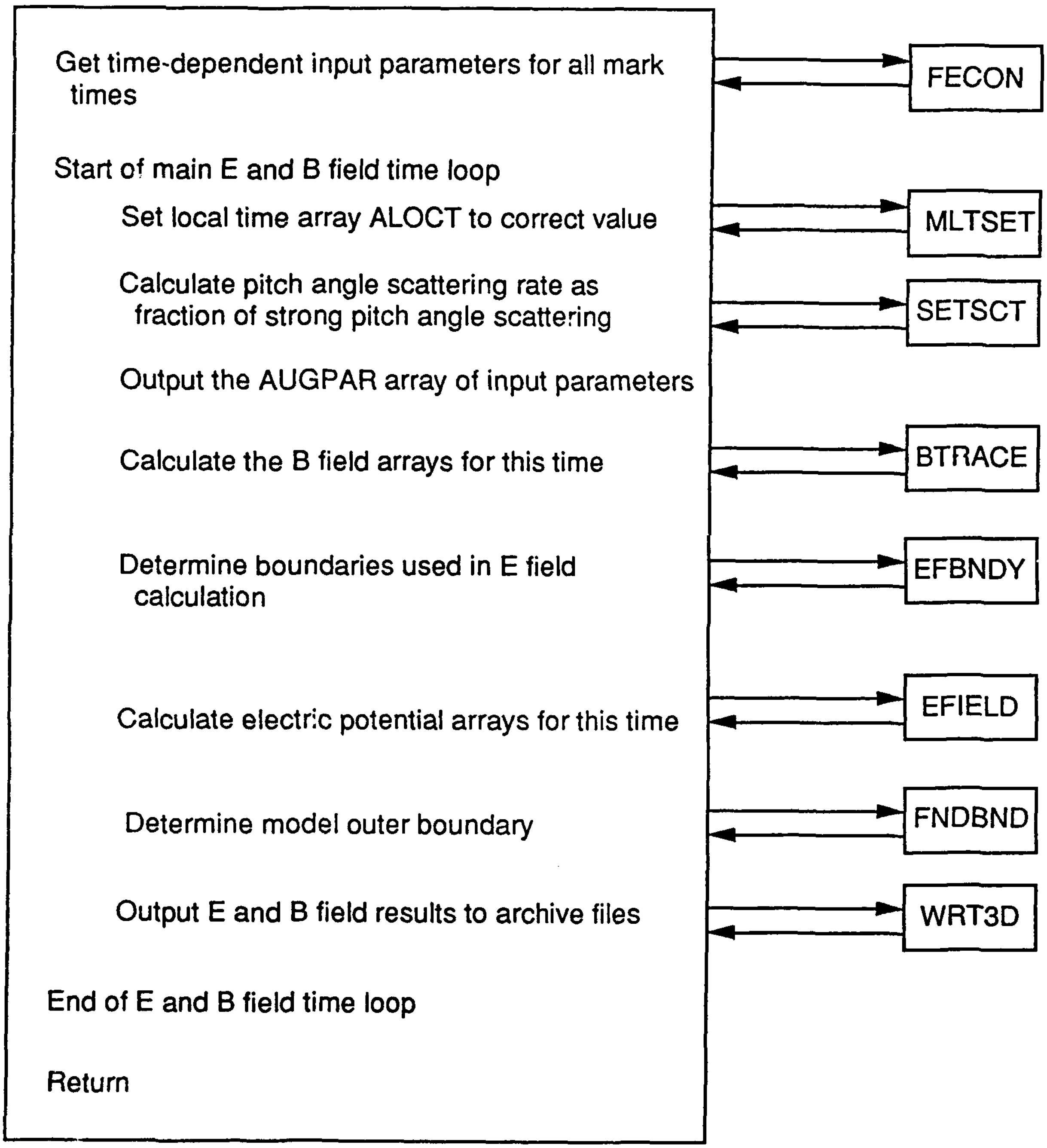
PPTCON

Subroutine that controls the major computations of the MSM. This routine calls EBCON to construct E and B field matrices at mark times and then proceeds to trace particle trajectories from *each grid point for each mark time.*



EBCON

Calculate E and B field arrays at grid points for mark times



PPTM

Subroutine to trace particle trajectories from each grid point for each particle species back in time one major time step from time = TSTART to time = TSTOP. (TSTART > TSTOP)

Calculate reference flux array	SETREF
Loop over all particle species	
Loop over all grid points	
Determine normalized time BBT	TNORML
Set ETA to boundary value of eta for points outside of boundary	
Calculate locations and energies at TSTART	G3NTRP
Start adaptive Runge-Kutta time loop	
Move particle a time step and increment time	MOVER
Interpolate to find location, flux tube volume and energy at end of small RK time step	G3NTRP
Calculate loss rate this time step	
If time boundary is crossed (ie., T<TSTOP) exit Runge-Kutta time loop	TCHK
If spatial boundary is crossed exit Runge-Kutta time loop	BNDCHK
Else go to beginning of Runge-Kutta time loop	
End of Runge-Kutta time loop	
Calculate ETA at grid point from traceback and loss information. ETA is not allowed to drop below equilibrium flux value. Loss depends on threshold for strong pitch angle scattering.	
Calculate precipitating flux and energy	
End loop over grid points	
End loop over species	
Return	

SOFTWARE UNIT SPECIFICATION

Software Unit Specification

2.2 Software Unit Functions. See the attached pages for the subroutine names, functions and calling environment.

2.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

2.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database. No file interrogation takes place during the run.

4.3 Inputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day)
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day)
IEND(2)	End Time (seconds)
IINC(1)	Program time increment (year)
IINC(2)	Program time increment (day)
IINC(3)	Program time increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following is the input files required, including those to be replace by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.

COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

4.3 Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume ((Re/nT)$^{-2/3}$). Output volume - number of latitudinal gridpoints × number of longitudinal

gridpoints × number of time steps × 4 (bytes)

BMIN — Equatorial magnetic field strength (nT).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

XMIN — GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

YMIN — GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

ZMIN — GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

BNDLOC — Location of outer boundary of detailed particle traces.
Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)

EFLUX — Precipitating energy flux array (ergs/cm^2-sec).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

FLXSUM — Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) (ergs/cm^2-sec).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)

EAVG — Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)

AUGPAR — Augmented data array for input values.
Output volume -number of data elements × number of time steps × 4 (bytes)

COLAT — Grid colatitude array (radians).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

ALOCT — Grid local time array (radians eastward from noon).
Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

FLUX — Flux values for all energy channels at all grid points

Output volume - number of latitudinal gridpoints $\times$ number of longitudinal gridpoints $\times$ number of time steps $\times$ number of energy channels $\times$ 4 (bytes)

MSM Routines by Function

Control Routines

Routine	*Purpose*	*Called by*	*Calls*
MSMCON	Main program Sets start and ending times and time step for run.	not applicable	PPTCON HIEPAR
PPTCON	Executive routine to control flow of MSM. E and B field models are computed in EBCON; particle traces are done in PPTM.	MSMCON	RDGRID OUTP FLXNIT DTIME EBCON EPREAD SETALM BNDSET RDHDR INITAL PPTM FLXDFL AURL1 PWRCAL WRT3D
EBCON	Executive routine to control calculation of electric and magnetic field models.	PPTCON	FECON MLTSET SETSCT BTRACE EFBNDY EFIELD OUTP FNDBND G3NTRP VMLSET WRT3D
HIEPAR	Subroutine to give energetic particle results.	MSMCON	presently dummy

Magnetic Field Routines

Routine	*Purpose*	*Called by*	*Calls*
BTRACE	Subroutine to calculate B field models appropriate for geophysical conditions as specified by FSTOFF, FEQEDG, FDST, FCLPSE, and FTILT.	EBCON	GETMAT RMVBSH BFGYRO
BFGYRO	Subroutine to rotate fixed grid B field arrays to match rotating coordinate system.	BTRACE	none
FNDBRK	Subroutine to find the proper B field array indices that bracket the magnetic field input parameters.	GETMAT	none
GETMAT	Subroutine to find the B field matrices needed for interpolation in order to represent current conditions.	BTRACE	MXINIT FNDBRK LOADBM ZEROBM OUTP
LOADBM	Subroutine to load individual B-matrices from the B-supermatrix into the work B-matrices.	GETMAT	none
MXINIT	Subroutine to initialize the array MEXIST to reflect the B-matrices that exist in the B-super matrix.	GETMAT	none
RMVBSH	Subroutine to extrapolate B-field matrices "into the bushes", ie., into extremely stretched or open regions.	BTRACE	none
ZEROBM	Subroutine to zero out an individual B matrix within the working B matrices. This is done when an individual matrix needed in the interpolation does not exist in the computed B-supermatrix.	GETMAT	none

Electric Field Routines

Routine	*Purpose*	*Called by*	*Calls*
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V).	EBCON	EMODEL
AURORA	Returns potential at a specific location in auroral region.	EMODEL	none
EFBNDY	Subroutine to calculate latitudes and widths of efield bndys 2 and 3.	EBCON	EFLOC G3NTRP
EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT).	EFBNDY	PFIX
EMODEL	Returns the electric potential arrays VNORTH, VSOUTH, and V at all grid pts.	EFIELD	INPUT THET EPOT LOW AURORA REG1
EPOT	Using scaled Heppner-Maynard model returns potential at a specific pt.	EMODEL	none
FNDBND	Subroutine to compute time dependent outer boundary location.	EBCON	THET
INPUT	Get input needed for computation of Heppner-Maynard model.	EMODEL	none
LOW	Returns potential at a specific location in low latitude region.	EMODEL	none
REG1	Returns potential at a specific location in region 1.	EMODEL	none
THET	Function that gives eqn for an E field boundary ellipse in flat polar coordinates.	EMODEL FNDBND PWRCAL	none

Particle Trace Routines

Routine	*Purpose*	*Called by*	*Calls*
PPTM	Subroutine to trace particle trajectories from each grid pt for each particle species back in time from time=TSTART to time=TSTOP.	PPTCON	SETREF OUTP TNORML G3NTRP WKRATE MOVER MODFIX TCHK BNDCHK FLXVAL
BNDCHK	Subroutine to determine whether particle has crossed bndy of calculation.	PPTM	TNORML
BNDSET	Subroutine to compute boundary plasma distribution.	PPTCON	none
CFACT	Function subprogram to compute correction factor based on equilibrium eta values.	not called	none
DVEFDI	Function subprogram to compute the I derivative of the effective potential at grid pt (I,J) at normalized time BT.	VLOCTY	none
DVEFDJ	Function subprogram to compute the J derivative of the effective potential at grid pt (I,J) at normalized time BT.	VLOCTY	none
FLXCAL	Subroutine to compute Kp-dependent flux at L=3,4,6.6, and 13 for electrons of given energy for a given Kp condition.	FLXNIT	none
FLXNIT	Subroutine to calculat Kp based particle number flux in units of particles/cm**2/s/sr/eV as function of r, energy, and Kp.	PPTCON	FLXCAL THRCAL
FLXVAL	Function subprogram to calculate Kp-dependent flux value at R=RR for particles of energy ENRG and KP=FKP by interpolating FLXMAT array.	PPTM BNDSET INITAL SETREF FLXDFL	G3NTRP

Routine	*Purpose*	*Called by*	*Calls*
FLXTRP	Function subprogram to return LOG10 of empirical flux value interpolated between values at L=3,4,6.6, and 13 Re.	FLXNIT	none
INITAL	Subroutine to compute initial plasma distribution	PPTCON	FLXVAL
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction.	PPTM	RK4
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values.	PPTCON	THET
RK4	4th order Runge-Kutta routine	MOVER	VLOCTY
RLYONS	Function subprogram to evaluate Lyons-based weak loss rate.	WKRATE	none
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program.	PPTCON	none
SETREF	Subroutine to set upper limit reference flux.	PPTM	FLXVAL TNORML
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency.	EBCON	none
TCHK	Subroutine to check whether time boundary was crossed.	PPTM	G3NTRP
THRCAL	Subroutine to calculate threshold flux at R=3, 4, 6.6, and 13 RE for KP=FKP and LOG(energy)=FNRGLG.	FLXNIT	none
VLOCTY	Subroutine to compute I and J components of particle velocity with energy invariant AALAM at location (BBI,BBJ) at normalized time BT.	RK4	G3NTRP

Routine	*Purpose*	*Called by*	*Calls*
VMLSET	Subroutine to calculate VMLOSS array.	EBCON	none
WKRATE	Function subprogram to evaluate weak precipitation loss rate.	PPTM	RLYONS

Input and Output Data Handling

Routine	*Purpose*	*Called by*	*Calls*
FECON	Subroutine to obtain time-normalized observational values.	EBCON	PARGEN
PARGEN	Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time.	FECON	INDATA DFLTVL SMOOTH DTXIPT DTNTRP TIMINC
DTNTRP	Subroutine to return interpolation values for the N data pairs of (XA(I),YA(I)), where I=1,NA.	PARGEN	none
DTXIPT	Subroutine to return interpolation values for XIPATT.	PARGEN	none
INDATA	Subroutine to read in data for the MSM during development and testing. This routine will be replaced in the operational MSM with an interface routine to the AWS environmental database.	PARGEN	none
OUTPUT	Subroutine to output MSM results to data base.	MSMCON	presently a dummy routine.
SMOOTH	Subroutine to check and extract data from DARRAY into XA and YA.	PARGEN	TCONV2
STNDOF	Subroutine to calculate the stand-off distance accordint to a formula from Alpbach (1979).	FECON	none

Default and Front-End Models

Routine	Purpose	Called by	Calls
ADDHDY	Subroutine to calculate the Hardy model electron flux value for a given point.	AURL1	FSUM
AURL1	Subroutine to determine default Hardy precipitating electron flux.	PPTCON	FNDHDY ADDHDY
AURL2S	Subroutine to calculate the default Hardy precipitating ion model.	PPTCON	REGEN EFUN
CLPDFL	Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model.	FECON	none
DSTDFL	Subroutine to calculate default (Kp driven) Dst value.	FECON	none
EFUN	Function subprogram used in evaluation of precipitating ion flux.	AURLS2	none
EQTDFL	Subroutine to calculate default (Kp driven) equatorward edge of auroral zone value.	FECON	none
FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid pts by interpolating the empirical FLXMAT array. This routine is used as the MSM default model when full particle traces are not done.	PPTCON	FLXVAL
FNDHDY	Subroutine to find and read the pertinent Fourier coefficients for the Hardy electron model.	AURL1	none
FSUM	Function subprogram that determines the sum of a Fourier series for Hardy model.	ADDHDY	none
PATDFL	Subroutine to return polar-cap convection pattern type on the basis of measurements of IMF By and Bz components if direct measurements of pattern type from DMSP are not available.	FECON	none

Routine	*Purpose*	*Called by*	*Calls*
PCPDFL	Subroutine to calculate default (Kp driven) cross-polar-cap potential value.	FECON	none
REGEN	Subroutine used in evaluation of Hardy model precipitating ion flux.	AURLS2	none
STNDFL	Subroutine to calculate the standoff distance. If solar wind velocity and density data are available, then Subroutine STNDOF does the actual calculation; otherwise, the default is Kp-based.	FECON	STNDOF
TILT	Subroutine to calculate the Earth's tilt angle. Tilt presently defaults to 0.	FECON	none

Utility Routines

Routine	*Purpose*	*Called by*	*Calls*
BLOCK DATA	Sets index of data elements in the PARRAY and AUGPAR arrays. Sets I,J, and energy indices for locations and species to be traced.	Not applicable	Not applicable
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at pt (BI,BJ,BK).	EBCON PPTM TCHK VLOCTY EFBNDY FLXVAL PWRCAL	none
G3TRPA	Utility function subprogram to perform a general 3-d linear interpolation of an angular array.	PPTM	none
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid pts.	EBCON	none
MODFIX	Subroutine to adjust BBJ to be between 2 and JDIM-1.	PPTM VLOCTY	none
OUTP	Utility subroutine to print out array information.	Utility routine	none
PFIX	Subroutine to adjust modulus of periodic J coordinate.	EFLOC	none
RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities.	PPTCON	none
RDHDR	Subroutine to read the header record of the standard MSM disk file format.	PPTCON	none
READ3D	Subroutine to read a record from the standard MSM disk file format.	PPTCON	none
TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.	SMOOTH	none

Routine	*Purpose*	*Called by*	*Calls*
TCONV3	Function to take a start year, start day, current year, current day, and current seconds of day to produce a value that is the number of seconds from midnight of the start year and day.	PARGEN	none
TIMINC	Subroutine to increment time for next E and B field record.	EBCON PARGEN FECON	none
TNORML	Utility function subprogram to calculate run-normalized time.	PPTM BNDCHK RK4 FNDLOC SETREF	none
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format.	PPTCON EBCON	none

USERS MANUAL

Users Manual

2.5 System Performance. Some factors on which the system performance depends include:

1. Higher energy particles require more time to process. The particles move so rapidly that a smaller time step is required to move their position and the number of time steps required to move the particles on the grid is correspondingly higher.

2. The output files are updated every program time step. If the program is set to run for many time steps, the storage of these files could be a problem. The maximum number of time steps is set within the program to 50.

3. The maximum number of energy channels which can be run is set within the program to 30. The greater the number of channels selected, the longer the run time will be.

2.7 Database/Data Bank. During development the method of input and output is through disk files. These files will no longer be necessary when the Environmental Data Base interface is in place. Attached is a list of the input parameters and units. Also attached is the proposed data transfer method.

2.8 General Description of Inputs, Processing, Outputs.

a. Inputs -

The following parameters are set within the program:

IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time increment (seconds)

Following are the input files required, including those to be replaced eventually by calls to the environmental data base:

* denotes the principal operator interfaces to the program

BOxxxxxx	The magnetic field matrices needed for the model run.

CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DKTABLE	Loss lifetimes for computing ion loss by charge exchange.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete.
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN*	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete.
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
MSMIN*	Specify start and stop times, a character list identifier, and sunspot number
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete.
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete.
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete.
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

b. Processing - After the data is gathered from the input files, it goes through Subroutine PARGEN which interpolates or extratpolates data as appropriate to provide data a a normalized run-time. Subroutine FECON then calculates the physical parameters needed by the model for each run time label. When the Environmental Database interface is operational, output from the model will then return to the Environmental Data Base through Subroutine OUTPUT. (At present OUTPUT) is only a place-holding stub.) The MSM Control and Data Flow Diagram is attached.

c. Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output for each time label.
VNORTH	Northern hemisphere electric potential distribution (Volts). Output for each time label.
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output for each time label
VM	Flux tube volume ($(Re/nT)^{-2/3}$). Output for each time label
BMIN	Equatorial magnetic field strength (nT), Output for each time label
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
BNDLOC	Location of outer boundary of detailed particle traces. Output for each time label
EFLUX	Precipitation energy flux array ($ergs/cm^2$-sec). Output for each time label
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) $O^{+)}$ ($ergs/cm^2$-sec). Output for each time label
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV). Output for each time label
AUGPAR	Augmented data array for input values. Output for each time label
COLAT	Grid colatitude array (radians). Output for each time label
ALOCT	Grid local time array (radians eastward from noon). Output for each time label
FLUX	Flux values for all energy channels at all grid points. Output for each time label for each energy channel

3.2 Input Requirements.

Start and stop times are specified to the program through the 5 line MSMIN file as year (eg., 1988), day (eg., 112 for the 112 th day of the year), and time in seconds from the beginning of the start day (eg., 43200 for noon). For example, the following configuration of the MSMIN file will run the

MSM from 1800 on the 112th day of 1988 through 1830 on the same day. This is a fresh startup of the program (direct access record number is zero) and the sunspot number is 0.0.

1988 112 64800	'start year, start day, start time in seconds'
1988 112 66600	'end year, end day, end time in seconds'
0	' 0 signifying new start'
'30 minute test run'	'up to 80 character run identification'
0.0	' sunspot number'

The time step increment is set in the program through the IINC vector.

IINC(1) = Increment Year
IINC(2) = Increment Day
IINC(3) = Increment Time (seconds)

If the run is to be a restarted using previous data, the direct access record number for restarting is also read from the MSMIN file. For example, the program is currently set up to run year 1988, day 112, hour 18 (64800 seconds) through year 1988, day 112, hour 1830 (66000 seconds) in 15 minute (900 second) increments. Since it is a new run the start record number is set to zero in MSMIN.

If after running this interval you want to restart the model at hour 1830 and continue through hour 20, the MSMIN input file would be:

1988 112 66600
1988 112 72000
3
'MSM run restart from 1830 on day 112'
0.0

The model would go back, pick up the original 5 records and continue the run from hour 19 to hour 20.

Input file ENCHAN controls how many and which energy channels are calculated. The file set up is:

Number of energy channels to be modeled (maximum 30,integer)
Particle type 1 (integer), Energy 1 (floating point)
Particle type 2 (integer), Energy 2 (floating point)
. . . .
Particle type n (integer), Energy n (floating point)

The acceptable particle type input is:

1	electrons
2	H^+
3	O^+

The energy is at geosynchronous orbit in keV. The energies must be in ascending order, with particle types grouped together, which are checked within the program.

For example,

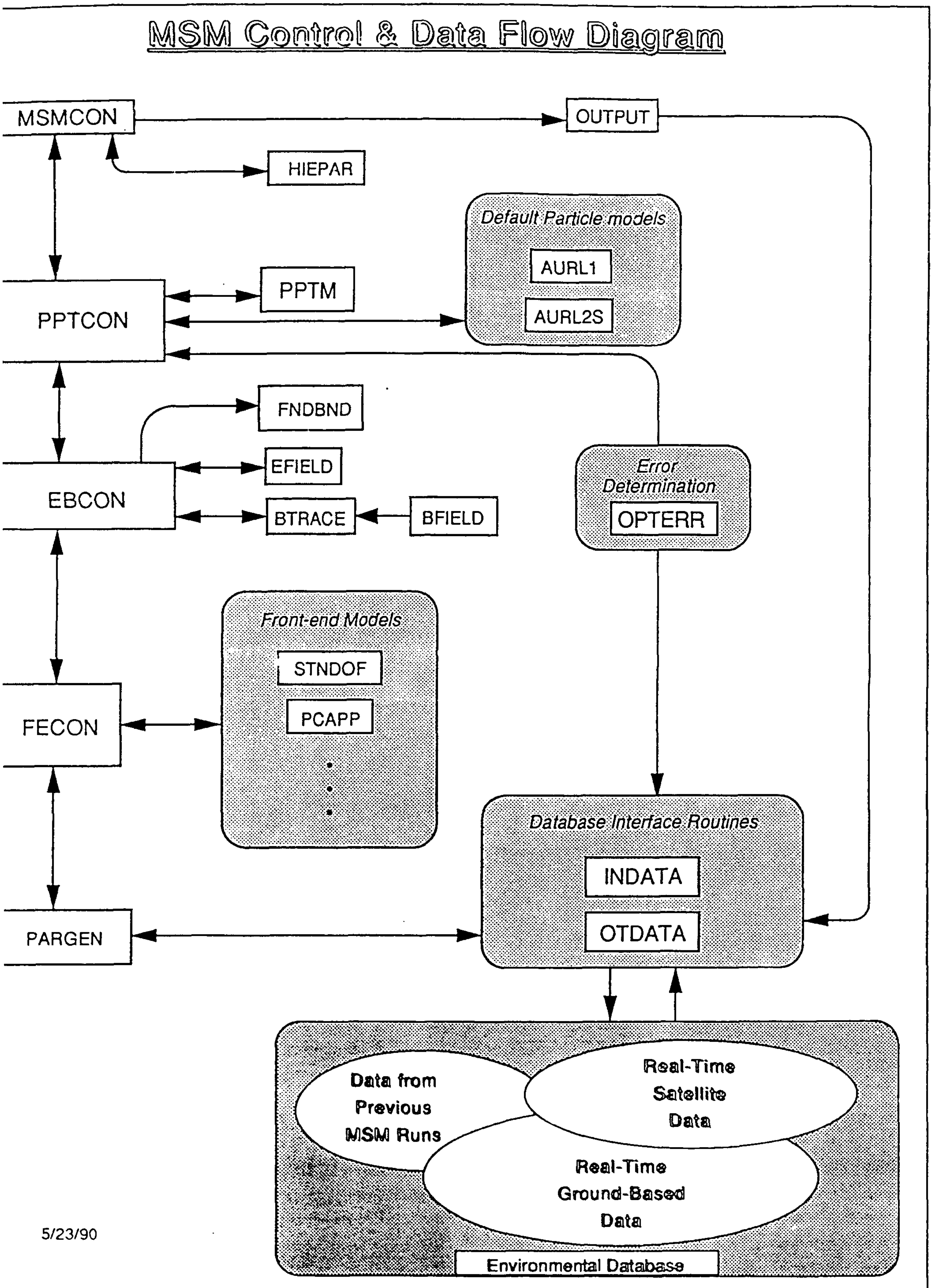
```
4
1 40
1 55
2 10
2 15
```

This is to model 4 energy channels, the first two are electrons of 40 and 55 keV at geosynchronous orbit. The second two are H^+ ions of 10 and 15 keV energy at geosynchronous orbit.

3.3 Output Requirements. The following files are output. When the Environmental Data Base is complete, these files will be stored there. The file of immediate use to forecasters is the FLUX data file which contains the particle flux information for the desired energy channels at all points on the model grid. The other data files will be needed for post-analysis of the event.

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output for each time label.
VNORTH	Northern hemisphere electric potential distribution (Volts). Output for each time label.
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output for each time label
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output for each time label
BMIN	Equatorial magnetic field strength (nT), Output for each time label
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
BNDLOC	Location of outer boundary of detailed particle traces. Output for each time label
EFLUX	Precipitation energy flux array ($ergs/cm^2$-sec). Output for each time label
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) $O^{+)}$ ($ergs/cm^2$-sec). Output for each time label
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV). Output for each time label

AUGPAR	Augmented data array for input values. Output for each time label
COLAT	Grid colatitude array (radians). Output for each time label
ALOCT	Grid local time array (radians eastward from noon). Output for each time label
FLUX	Flux values for all energy channels at all grid points. Output for each time label for each energy channel


MSM Control & Data Flow Diagram
MSMCON
OUTPUT
HIEPAR
Default Particle models
AURL1
AURL2S
PPTM
PPTCON
FNDBND
EFIELD
EBCON
BTRACE
BFIELD
Error Determination
OPTERR
Front-end Models
STNDOF
PCAPP
FECON
Database Interface Routines
INDATA
OTDATA
PARGEN
Data from Previous MSM Runs
Real-Time Satellite Data
Real-Time Ground-Based Data
Environmental Database
5/23/90

PARAMETER	NAME	UNITS	I/O	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
INPUT ONLY:								
Kp	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(I)	EFLX(I)	$^{ELEC}/_{CM^2\text{-}S}$	I	X				X
GEO ION FLUX - CHANNEL(I)	GFLX(I)	$^{IONS}/_{CM^2\text{-}S}$	I	X				X
SOLAR WIND VELOCITY	SWVEL	$^{KM}/_{S}$	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	$^{PROT}/_{CM^3}$	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFBY	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

END USERS MANUAL

End Users Manual

2.1.2 Performance. Some factors on which the system performance depends include:

1. Higher energy particles require more time to process. The particles move so rapidly that a smaller time step is required to move their position and the number of time steps required to move the particles on the grid is correspondingly higher.

2. The output files are updated every program time step. If the program is set to run for many time steps, the storage of these files could be a problem. The maximum number of time steps is set within the program to 50.

3. The maximum number of energy channels which can be run is set within the program to 30. The greater the number of channels selected, the longer the run time will be.

COMPUTER OPERATION MANUAL

Computer Operation Manual

2.1 System Application. The Magnetospheric Specification Model was developed as a set of algorithms to be developed into a real-time program for use in the Space Forecast Center operated by the Air Weather Service of the U.S. Air Force. It is intended to help the Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.
2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.4 Information Inventory.

2.4.1 Resource Inventory.

Following are the files accessed by the MSM, including those to be eventually replaced by calls to the environmental data base. Those required for a restart are so labelled.

Input data sets

* denotes principal mode of operator control of program

BOxxxxxx	The magnetic field matrices needed for the model run. Required for restart.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system. Required for restart.
DKTABLE	File of decay times for computing ion loss by charge exchange.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model. Required for restart.
ENCHAN*	Input values for the number of energy channels, type and energy of particles to be traced. Required for restart.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete. Required for restart.
HARDY	Coefficients used by the Hardy precipitating electron model. Required for restart.
IONENG	Coefficients used as input for the ion precipitating model. Required for restart.

IONNUM	Coefficients used as input for the ion precipitating model. Required for restart.
MSMIN*	Primary input file to control run. Provides start and end times for run, a character string identifier for run, and sun spot number. Required for restart.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Output data sets

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Equatorial magnetic field strength (nT). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).

	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces.
	Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)
EFLUX	Precipitating energy flux array (ergs/cm^2-sec).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) (ergs/cm^2-sec).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
AUGPAR	Augmented data array for input values.
	Output volume -number of data elements × number of time steps × 4 (bytes)
COLAT	Grid colatitude array (radians).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ALOCT	Grid local time array (radians eastward from noon).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
FLUX	Flux values for all energy channels at all grid points. Needed for restart.
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

2.4.2 Report Inventory. A print file of arrays of different parameters is currently generated.

TEST PLAN

Test Plan

2. DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity. During the development phase the model output was compared to observed data for 2 events - CDAW6 March 22, 1978 interval and April ~~22-24~~ 21-23, 1988 interval.

2.2 Pretest Activity Results. Results of the CDAW6 interval are discussed in "Chapter IV - Testing the Model Using the CDAW6 Event" of B. Hausman's Masters Thesis (included as an appendix to this test plan.) MSM results are compared quantitatively with observations for the April ~~22-24~~ 21-23, 1988 period in the Rice MSM final report.

3.1 System Description.

Start and stop times and start direct-access record number are provided in input file MSMIN.

Following are the input files required, including those to be replace by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DKTABLE	Decay times for computing charg-exchange loss of ions
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.

PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Following are the output files:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume $((Re/nT)^{-2/3})$.
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm^2-sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) $O^{+)}$ (ergs/cm^2-sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOCT	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

4. TEST SPECIFICATION AND EVALUATION

4.1 Test Specification. The set of algorithms encoded as the MSM have been run and tested at Rice University. The results have been studied by comparing the model results with observed data. The version of the program code which was developed at Rice University gives the verified results. To validate the algorithms delivered to the Space Forecast Center, the output from the code run at Rice University will be compared with that run at the Space Forecast Center.

4.2 Test Methods and Constraints.

4.2.1 Test Conditions. The input data will come from Rice University.

4.4 Test Evaluation.

4.4.1 Test Data Criteria. Using the output data furnished from Rice University, the output from the code run at the Space Forecast Center should be the same.

IV. TESTING THE MODEL USING THE CDAW6 EVENT

A. General Description of Event

The Coordinated Data Analysis Workshop #6 (CDAW6) focused its effort on understanding the physical processes which control the flow of energy from the solar wind through the magnetosphere and the eventual release through magnetospheric substorms. The participants studied the first of 2 large substorms which occurred on March 22,1979. Preceded by a day of quiet conditions both in the solar wind and the magnetosphere, a storm sudden commencement was observed at 0826UT followed by the onset of the expansive phase of the first substorm at 1054UT. The ready availability of a large data set and numerous papers (see Journal of Geophysical Research volume 90, February 1985 for most of them) made it a natural candidate for the preliminary testing of the model's capabilities.

Figure 4.1 is a summary of solar wind plasma parameters at IMP-8 (located slightly to the duskside of the bow shock) and the AE magnetic activity index published as Figure 2 in the overview paper by McPherron and Manka (1985). The major features of these data include:

1) An interplanetary shock arrives at IMP-8 at 0821:20UT, as indicated by an increase in solar wind density, velocity and magnetic field strength.

2) At 0826UT the shock arrives at the dayside of the earth compressing the magnetosphere causing a sudden storm commencement, which can be seen as a small spike in the AE index.

3) The magnetic field turns southward at IMP-8 at 1008UT which begins the stretching of the magnetotail and the growth phase of the first substorm.

4) The first substorm expansion onset at 1054UT produced auroral zone magnetic activity (as measured by the AE index) exceeding 1000 nT.

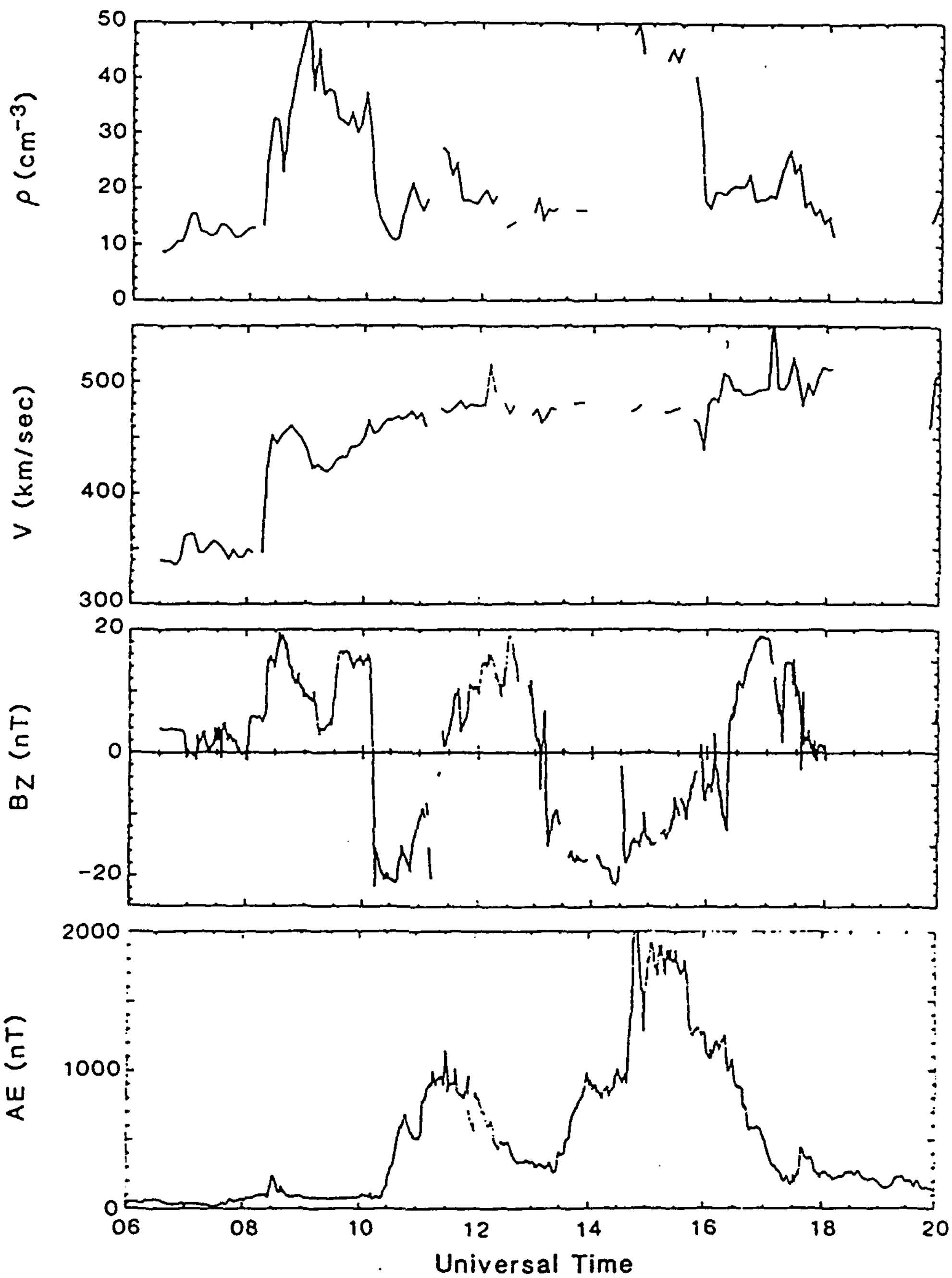


Figure 4.1 Summary of solar wind plasma parameters at IMP 8 on March 22,1979. Top traces show 5-min averages of solar wind velocity and density. Middle trace shows 1-min values of the GSM Z component of the IMP at IMP 8. Bottom trace presents a 55-station AE index calculated at 1 min time resolution.

5) At 1122UT the IMF becomes northward at IMP-8, magnetic activity starts to decline and the recovery phase of the substorm begins.

B. Model Inputs

1. Magnetic Field Model

After studying the IMF data and Dst index, Dr. G.-H. Voigt subdivided the CDAW-6 interval into 17 time labels (Table 4.1) which represent different configurations of the magnetic field during the event. The magnetic field matrices were then pre-computed for each time. The inputs to the model are

STAND - the magnetopause standoff distance (Re) from the solar wind data.

Dst - the standard index of ring current intensity (nT).

Aur.Eq.Bd. - the equatorward edge of the auroral zone at midnight (information on how this is done is in Section 2 - Electric Field Model Inputs).

Map(E) - the expected mapping from the Aur.Eq.Bd. (above) to the inner edge of the plasma sheet at local midnight in the equatorial plane of the magnetosphere (Re). This was estimated using the work by Kivelson et al.(1979), Horowitz et al. (1986), and Gussenhoven et al. (1981) who have calculated Kp-based relationships between the inner edge of the plasma sheet and the equatorward edge of the auroral zone.

From these four observable parameters, Hilmer (1989) has devised a scheme to estimate the remaining input values necessary to run the model:

TILT - Dipole tilt angle (set to 0.0 for the CDAW-6 event).

HJNEAR - Strength of the cross-tail current sheet at the inner edge.

XNEAR - Location of the inner edge of the model plasma sheet at local midnight (Re).

HJFRAC - Fraction of HJNEAR appearing at and beyond (XNEAR-100) Re. Current intensity is constant in the far tail.

LABEL = 1	6:00	UT	Pre-Storm Sudden Commencement
LABEL = 2	8:26	UT	Storm Sudden Commencement
LABEL = 3	10:00	UT	Same as LABEL 2 but little bit more stretch
LABEL = 4	10:20	UT	IMF turns southward
LABEL = 5	10:54 - EPS		Maximum stretch just before onset of first substorm
LABEL = 6	10:54 + EPS		Collapse right after onset of substorm
LABEL = 7	11:30	UT	End of phase "gradually back to normal"
LABEL = 8	12:06	UT	Nominal phase (i.e. nominal model)
LABEL = 9	12:43	UT	Nominal phase
LABEL = 10	13:10	UT	IMF turns southward
LABEL = 11	13:40	UT	Gradual stretching period
LABEL = 12	14:07	UT	Gradual stretching period
LABEL = 13	14:36 - EPS		Maximum stretch just before onset of second substorm
LABEL = 14	14:36 + EPS		Collapse right after onset of substorm
LABEL = 15	15:15	UT	Gradually back to normal after collapse
LABEL = 16	15:50	UT	Gradually back to normal
LABEL = 17	16:00	UT	Beginning of last nominal phase

Table 4.1 Critical time labels for CDAW-6 event.

DY - Scale length for changes in tail structure along the GSM y-axis (Re). This applies to the westward tail current system.

DFIL - Plasma sheet thickness parameter.

BRN, BRP - Ring-current contributions (nT) at Earth's center from the westward and eastward components, respectively (BRN < 0, BRP > 0).

RN, RP - Characteristic radii of the westward and eastward ring-current components, respectively.

COLLAP - Fraction of the pre-collapse midnight cross-tail current remaining in the disturbed region (from the inner edge, XNEAR to XNEAR - DELXC). COLLAP = 1.0 means no collapse.

DELXC - The distance from the inner edge (Re), along the x-axis, which is altered by the addition of eastward tail current. DELXC = 0.0 means no collapse.

DYC - Scale length for changes in tail structure along the GSM y-axis (Re). Applies to the eastward disruption collapse current. DYC is always less than DY. DYC = 0.0 means no collapse.

Table 4.2 from Dr. R. Hilmer contains a list of these parameters for the CDAW-6 event. Also listed are three output parameters:

MAP(R) - The resulting midnight equatorial mapping distance (Re).

RC Zero - Characteristic radius for the ring current (Re). The ring current changes sign at this distance and is an implied particle pressure maximum.

Min-Loc - The minimum |B| and its location in the equatorial plane at midnight.

2. Electric Field Model

The cross polar cap potential was calculated from the IMF data and the empirical formula suggested by Crooker et al. (1982) and presented by Reiff and Luhmann (1986)

$$\phi(\mathrm{kV}) = 22 + 0.069\ vB\ Q1$$

where $Q1 = \alpha(\alpha - \cos\theta)\,(\alpha^2 + 1 - 2\alpha\cos\theta)^{-1/2}$ for $\cos\theta < \alpha$

	Label	UT	Comment	Stand	Dst	Aur.Eq.Bd.	Map(E)	Map(R)	RC Zero	Min-Loc
1	1	6:00	Pre-SSC	10.58	13.53	66.28	7.01	7.00	3.56	0.72 - 36
2	2	8:26	SSC	7.83	43.20	66.28	7.01	7.01	3.21	0.81 - 26
3	3	10:00	"	7.83	43.20	65.25	6.69	6.69	3.21	1.70 - 20
4	4	10:20	IMF South	8.93	32.56	64.72	6.53	6.52	3.20	1.24 - 20
5	5	10:54	Stretched	8.45	16.61	62.51	5.86	5.83	3.04	0.50 - 17
6	6	10:54+	Collapse	8.45	16.61	62.51			3.04	0.42 - 24
7	7	11:30	Nominal	8.10	-15.30	59.22	4.38	4.33	2.51	3.16 - 21
8	8	12:06	"	8.45	-24.41	58.82	4.24	4.23	2.41	3.21 - 21
9	9	12:43	"	8.77	-18.34	59.77	4.58	4.54	2.60	2.80 - 21
10	10	13:10	IMF South	8.71	-13.78	60.72	4.92	4.94	2.66	2.48 - 20
11	11	13:40		8.64	-9.22	60.73	4.92	4.90	2.64	2.44 - 20
12	12	14:07		8.61	-16.82	60.73	4.92	4.92	2.56	2.11 - 18
13	13	14:36	Stretched	7.58	-22.14	60.13	4.71	4.72	2.42	2.90 - 20
14	14	14:36+	Collapse	7.58	-22.14	60.13			2.42	1.09 - 26
15	15	15:15	Nominal	7.62	-39.61	59.80	4.59	4.58	2.24	2.57 - 22
16	16	15:50	"	7.68	-57.08	59.27	4.40	4.40	2.14	1.49 - 26
17	17	16:00	"	8.50	-60.88	59.14	4.35	4.35	2.14	1.40 - 26

	HJNEAR	XNEAR	HJFRAC	DY	DFIL	COLLAP	DELXC	DYC	BRN	BRP	RN	RP
1	4.3	-7.00	0.25	15.0	2.0	1.0	0.0	0.0	-105.0	117.2	3.2	2.1
2	8.3	-7.01	0.20	12.5	2.0	1.0	0.0	0.0	-50.0	75.9	4.0	1.6
3	12.0	-6.69	0.15	10.0	1.2	1.0	0.0	0.0	-50.0	75.9	4.0	1.6
4	12.7	-6.53	0.15	10.0	1.0	1.0	0.0	0.0	-50.0	72.0	4.0	1.6
5	17.5	-5.86	0.14	7.5	0.8	1.0	0.0	0.0	-230.0	221.6	2.9	2.1
6	17.5	-5.86	0.14	7.5	0.8	0.7	45.0	5.0	-230.0	221.6	2.9	2.1
7	15.0	-4.38	0.15	7.5	1.0	1.0	0.0	0.0	-400.0	354.5	2.7	2.1
8	15.0	-4.24	0.15	7.5	1.0	1.0	0.0	0.0	-490.0	437.1	2.6	2.1
9	15.0	-4.58	0.15	7.5	1.0	1.0	0.0	0.0	-390.0	345.8	2.7	2.1
10	15.0	-4.92	0.15	7.5	1.0	1.0	0.0	0.0	-380.0	340.4	2.7	2.1
11	15.0	-4.92	0.15	7.5	1.0	1.0	0.0	0.0	-540.0	504.9	2.6	2.2
12	17.0	-4.92	0.14	7.5	1.0	1.0	0.0	0.0	-500.0	466.4	2.5	2.1
13	17.0	-4.71	0.14	7.5	1.0	1.0	0.0	0.0	-570.0	522.9	2.5	2.1
14	17.0	-4.71	0.14	7.5	1.0	0.8	35.0	5.0	-570.0	522.9	2.5	2.1
15	15.0	-4.59	0.15	7.5	2.0	1.0	0.0	0.0	-840.0	774.3	2.4	2.1
16	11.0	-4.40	0.20	10.0	2.0	1.0	0.0	0.0	-700.0	615.9	2.5	2.1
17	11.0	-4.35	0.20	10.0	2.0	1.0	0.0	0.0	-680.0	597.9	2.5	2.1

Table 4.2 Input parameters for B-field model for CDAW-6 interval.

$$Q1 = 0 \qquad\qquad \text{for } \cos\theta > \alpha$$

$$\alpha = \min\left[\frac{(4B_T B_E)^{1/2}}{60\ \mathrm{nT}}, 1\right]$$

θ = polar angle of the IMF

B_T = projection of IMF on solar-magnetosphere y-z plane

B_E = magnetic field just inside magnetopause

The polar cap pattern type was also estimated from By and Bz IMF data for each time label using the criteria given in chapter 3. For the CDAW-6 interval, the equatorward edge of the diffuse aurora was based on electron data from DMSP, P78, and TIROS satellites.

Table 4.3 contains the input data calculated by Dr. R. A. Wolf and used during the interval for:

VPVC - cross-polar cap potential (kV).

IPATTV - polar cap pattern type.

EDGE - equatorward edge of auroral zone at midnight.

A(L) - radius of ellipse measured in x (sunward) direction.

B(L) - radius of ellipse measured in y (duskward) direction.

DX(L) - sunward displacement of coordinate system center from pole.

DY(L) - duskward displacement of coordinate system center from pole.

3. Boundary Plasma Condition

The invariant temperatures (λ_{kTe}, λ_{kTi}) were estimated from data from the ISEE-2 satellite, as shown in Figure 4.2 and published as Figure 1 in Paschmann et al. (1985). The satellite moved earthward from ~14.8 to ~13.0 Re geocentric distance at ~0200 local time from 1010-1200 UT. However, the published observations do not cover the entire

	1	2	3	4	5	6	7	8
1	Time	VPCV	IPATTV	EDGE	A(1)	A(2)	A(3)	B(1)
2	0 - 0826-e	24	4	66.25	15.47	16.81	21.88	15.67
3	0826+e	24	5	66.25	17.85	19.25	21.88	18.52
4	1000	24	5	65.35	18.75	20.48	22.63	18.63
5	1020	66	3	64.80	18.70	20.66	23.10	17.56
6	1054-e	137	3	62.80	20.88	22.89	24.78	19.05
7	1054+e	137	3	62.80	19.70	21.61	24.78	18.78
8	1130	130	3	59.00	20.94	22.72	27.97	20.12
9	1206	107	3	59.00	20.91	22.71	27.97	19.86
10	1243	88	3	59.90	20.77	22.61	27.21	19.39
11	1310	94	3	60.20	20.72	22.59	26.96	19.31
12	1340	135	3	60.85	20.61	22.47	26.42	19.25
13	1407	161	3	60.60	20.76	22.68	26.63	19.17
14	1436-e	156	3	60.10	21.34	23.15	27.05	20.64
15	1436+e	156	3	60.10	20.74	22.49	27.05	20.49
16	1515	144	2	59.80	20.99	22.71	27.30	20.75
17	1550	127	2	59.25	20.73	22.31	27.76	21.08
18	1600	122	2	59.25	20.33	22.02	27.76	20.19

	9	10	11	12	13	14	15	16
1	B(2)	B(3)	DX(1)	DX(2)	DX(3)	DY(1)	DY(2)	DY(3)
2	14.04	22.83	-1.00	-0.56	-1.87	-0.90	0.00	0.00
3	17.09	22.83	-1.19	-0.81	-1.87	-0.77	0.00	0.00
4	17.40	23.74	-1.54	-1.62	-2.02	-0.64	0.00	0.00
5	16.53	24.30	-1.73	-2.21	-2.10	-0.64	0.00	0.00
6	17.82	26.33	-1.96	-2.48	-2.42	-0.64	0.00	0.00
7	17.50	26.33	-1.36	-1.78	-2.42	-0.64	0.00	0.00
8	18.88	30.18	-1.08	-1.37	-3.03	-0.64	0.00	0.00
9	18.53	30.18	-1.10	-1.40	-3.03	-0.64	0.00	0.00
10	18.04	29.27	-1.27	-1.62	-2.89	-0.64	0.00	0.00
11	17.97	28.96	-1.35	-1.73	-2.84	-0.64	0.00	0.00
12	17.90	28.30	-1.35	-1.73	-2.74	-0.64	0.00	0.00
13	17.79	28.56	-1.56	-1.99	-2.78	-0.64	0.00	0.00
14	19.46	29.06	-1.20	-1.53	-2.86	-0.64	0.00	0.00
15	19.28	29.06	-0.89	-1.15	-2.86	-0.64	0.00	0.00
16	19.49	29.37	-0.68	-0.92	-2.90	-0.64	0.00	0.00
17	20.07	29.92	-0.31	-0.40	-2.99	-0.64	0.00	0.00
18	19.06	29.92	-0.48	-1.62	-2.99	-0.64	0.00	0.00

Table 4.3 Input parameters for E-field model for CDAW-6 interval.

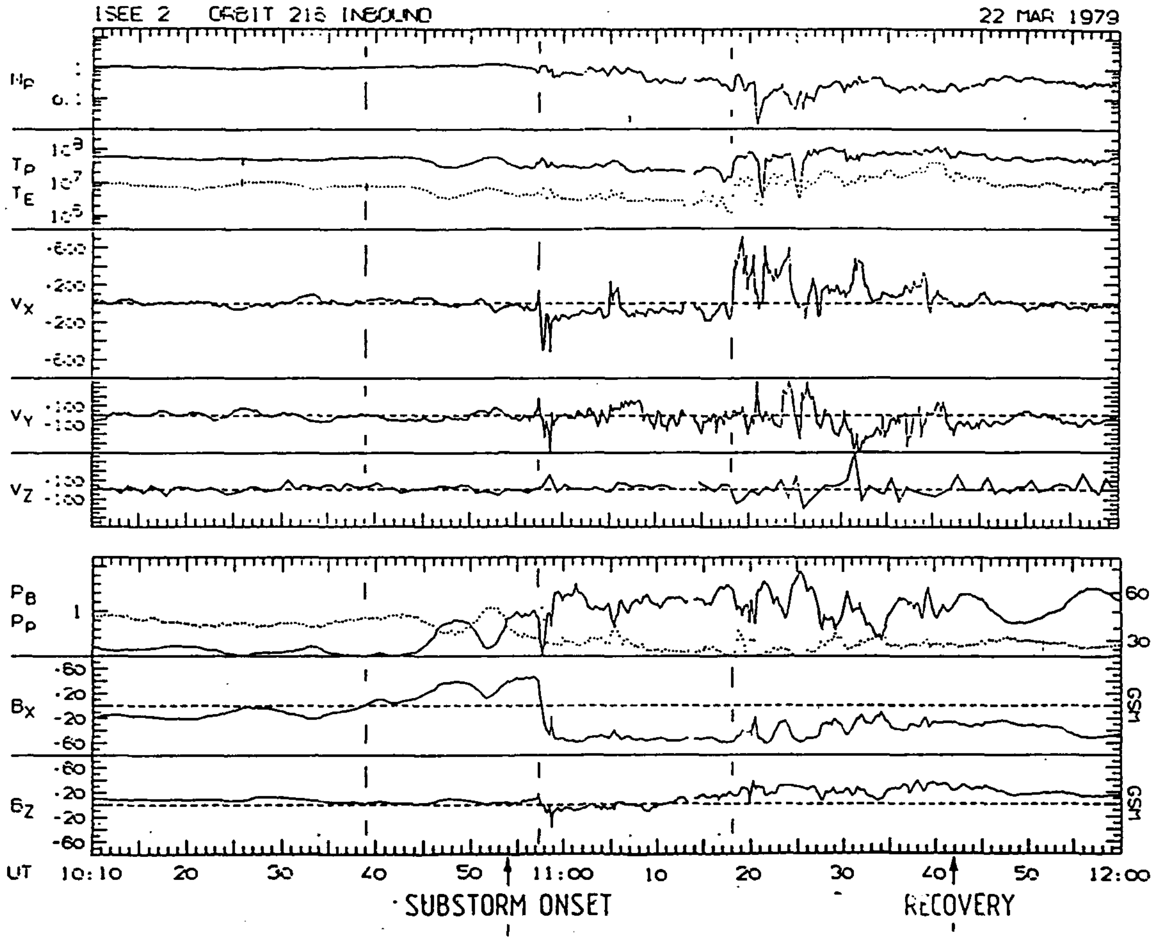


Figure 4.2 ISEE 2 plasma and magnetic field parameters for the time interval 1010-1200 UT on March 22, 1979. T_p and T_e are the ion and electron temperatures (degrees Kelvin).

time interval we want to model, so we have estimated the temperatures from the data and extrapolated to cover the rest of the event. The top of Table 4.4 shows the invariant temperatures calculated for the entire event. The headings are

Label - Time label of magnetic field model (See Table 4.1).

t_{figure} - Time of ISEE-2 data point used to calculate temperature.

r_{figure} - Radial distance of ISEE-2 (Re).

VM_{model} - Flux tube volume at r_{figure} from the magnetic field model.

T_i (°K) - Estimated ion temperature from ISEE-2 data.

T_e (°K) - Estimated electron temperature from ISEE-2 data.

λ_{kTi} - Calculated ion invariant temperature.

λ_{kTe} - Calculated electron invariant temperature.

$<|\lambda_i|>$ - average value of ion energy invariant ($= 1.5 \times \lambda_{kTi}$).

$<|\lambda_e|>$ - average value of electron energy invariant ($= 1.5 \times \lambda_{kTe}$).

The bottom figure shows the value of η_{tot} calculated from equation 3.24 in Chapter 3 at a distance of 20 Re and local midnight. The headings are

UT_{model} - the universal time of the model.

$B_{lobe}{}^2/ 8\pi$(model(20Re)) - lobe magnetic pressure at 20 Re and midnight.

VM_{model} (20Re) - flux tube volume at 20 Re and midnight.

η_{tot} - calculated total invariant density at 20 Re and midnight .

The initial invariant density for each energy channel is given by equation 3.31 in Chapter 3. A list of the energy channels (K), invariant energy (ALAM(K)), and boundary values at 20 Re and local midnight (ETAB(K)) is given in Table 4.5.

4. Initial Plasma Condition

The initial plasma condition is set using K=32, the highest electron energy that we model and it corresponds to ~ 40keV at geosynchronous orbit. Using the 30-45 keV electron channel of the1977-007 and 1976-059 spacecraft we estimated the average flux

Label	t_{figure}	r_{figure}	VM_{model}	T_i (°K)	T_e (°K)	λ_{kTi}	λ_{kTe}	$<\vert\lambda_i\vert>$	$<\vert\lambda_e\vert>$
1	10:26am	14.5	1.111	$6.4x10^7$	$1.0x10^7$	4960	775	7440	1162.5
2	10:26am	14.5	1.124	$6.4x10^7$	$1.0x10^7$	4908	767	7362	1150.5
3	10:26am	14.5	1.147	$6.4x10^7$	$1.0x10^7$	4810	752	7215	1128.
4	10:26am	14.5	1.121	$6.4x10^7$	$1.0x10^7$	4921	769	7381.5	1153.5
5	10:26am	14.5	1.453	$6.4x10^7$	$1.0x10^7$	3797	593	5695.5	889.5
6	10:26am	14.5	1.246	$6.4x10^7$	$1.0x10^7$	4428	692	6642	1038
7	11:50am	13.16	1.719	$1.0x10^8$	$2.0x10^7$	5015	1003	7522.5	1504.5
8	11:50am	13.16	1.730	$1.0x10^8$	$2.0x10^7$	4983	996	7474.5	1494
9	11:50am	13.16	1.649	$1.0x10^8$	$2.0x10^7$	5228	1045	7842	1567.5
10	10:26am	14.5	1.287	$6.4x10^7$	$1.0x10^7$	4287	670	6430.5	1005
11	10:26am	14.5	1.339	$6.4x10^7$	$1.0x10^7$	4120	644	6180	966
12	10:26am	14.5	1.316	$6.4x10^7$	$1.0x10^7$	4192	655	6288	982.5
13	10:26am	14.5	1.346	$6.4x10^7$	$1.0x10^7$	4099	640	6148.5	960
14	10:26am	14.5	1.296	$6.4x10^7$	$1.0x10^7$	4257	665	6385.5	997.5
15	11:50am	13.16	1.593	$1.0x10^8$	$2.0x10^7$	5412	1082	8118	1623
16	11:50am	13.16	1.642	$1.0x10^8$	$2.0x10^7$	5250	1050	7875	1575
17	11:50am	13.16	1.613	$1.0x10^8$	$2.0x10^7$	5345	1069	8017.5	1603.5

Label	UT_{model}	$B_{lobe}{}^2/8\pi$(model ($20R_E$))	$VM_{model}(20R_E)$	η_{tot}
1	0 - 0826-ε	$8.049x10^{-10}$	0.6049	$1.966x10^{21}$
2	0826+ε	$20.019x10^{-10}$	0.6050	$4.940x10^{21}$
3	1000	$33.054x10^{-10}$	0.7366	$5.088x10^{21}$
4	1020	$35.461x10^{-10}$	0.7335	$5.393x10^{21}$
5	1054-ε	$65.914x10^{-10}$	0.9300	$7.178x10^{21}$
6	1054+ε	$34.808x10^{-10}$	0.6462	$8.075x10^{21}$
7	1130	$47.003x10^{-10}$	0.8949	$4.111x10^{21}$
8	1206	$49.956x10^{-10}$	0.9148	$3.912x10^{21}$
9	1243	$46.720x10^{-10}$	0.8730	$4.170x10^{21}$
10	1310	$46.792x10^{-10}$	0.8709	$5.137x10^{21}$
11	1340	$46.601x10^{-10}$	0.8631	$5.636x10^{21}$
12	1407	$55.492x10^{-10}$	0.8923	$6.069x10^{21}$
13	1436-ε	$56.765x10^{-10}$	0.9280	$5.757x10^{21}$
14	1436+ε	$43.138x10^{-10}$	0.8166	$5.799x10^{21}$
15	1515	$45.457x10^{-10}$	0.7328	$6.071x10^{21}$
16	1550	$31.912x10^{-10}$	0.6988	$4.948x10^{21}$
17	1600	$31.526x10^{-10}$	0.6860	$5.028x10^{21}$

Table 4.4 Invariant temperature and total invariant density calculated from CDAW-6 interval.

K	ALAM(K)	ETAB(K)
1	374.80640	0.86136E+20
2	1202.52612	0.15263E+21
3	2146.50952	0.19207E+21
4	3223.09546	0.21585E+21
5	4450.89844	0.22572E+21
6	5851.17187	0.22246E+21
7	7448.13672	0.20738E+21
8	9269.42187	0.18272E+21
9	11346.53120	0.15164E+21
10	13715.41020	0.11792E+21
11	16417.03520	0.85332E+20
12	19498.16410	0.56979E+20
13	23012.07810	0.34758E+20
14	27019.57030	0.19146E+20
15	31589.98440	0.93944E+19
16	36802.38670	0.62709E+19
17	-58.56357	0.86136E+20
18	-187.89493	0.15263E+21
19	-335.39233	0.19207E+21
20	-503.60913	0.21585E+21
21	-695.45410	0.22572E+21
22	-914.24683	0.22246E+21
23	-1163.77295	0.20738E+21
24	-1448.34888	0.18272E+21
25	-1772.89771	0.15164E+21
26	-2143.03564	0.11792E+21
27	-2565.16455	0.85332E+20
28	-3046.59180	0.56979E+20
29	-3595.64160	0.34758E+20
30	-4221.81250	0.19146E+20
31	4935.93750	0.93945E+19
32	-5750.37891	0.62709E+19

Table 4.5 A list of the energy channels (K), invariant energy (ALAM(K)), and boundary values at 20 Re and local midnight (ETAB(K)).

to be $10^{1.48}$ $cm^{-2}s^{-1}sr^{-1}eV^{-1}$ (See Figure 4.3). The other parameters needed to solve equation 3.40 are:

ALAM(K) - energy invariant for energy K=32; = -5750.4 $(Re/nT)^{2/3}$

VM - magnetic flux tube volume at 6.6 Re on dawn/dusk meridian

= 6.82 $(Re/nT)^{-2/3}$

λ_{kmax} - defined in equation 3.33a in Chapter 3

= -7451.8 eV $(Re/nT)^{2/3}$

λ_{kmin} - defined in equation 3.32b in Chapter 3

= -5343.2 $(Re/nT)^{2/3}$

Solving for EETA(32), the invariant density at geosynchronous orbit is

EETA(32) = 1.666E17

The invariant density for K=32 at the back boundary is given in Table 4.4 as 0.6271E19. The initial back boundary at local midnight at 15.87 Re. Solving equation 3.42 in Chapter 3 to get the power law coefficient:

$$\frac{\text{EETA(Geosynchronous)}}{\text{EETA(Boundary)}} = \frac{1.666\text{E}17}{0.627\text{E}19} = 0.0266 = \left[\frac{6.6\ \text{Re}}{15.87\ \text{Re}}\right]^{p}$$

p = 4.134

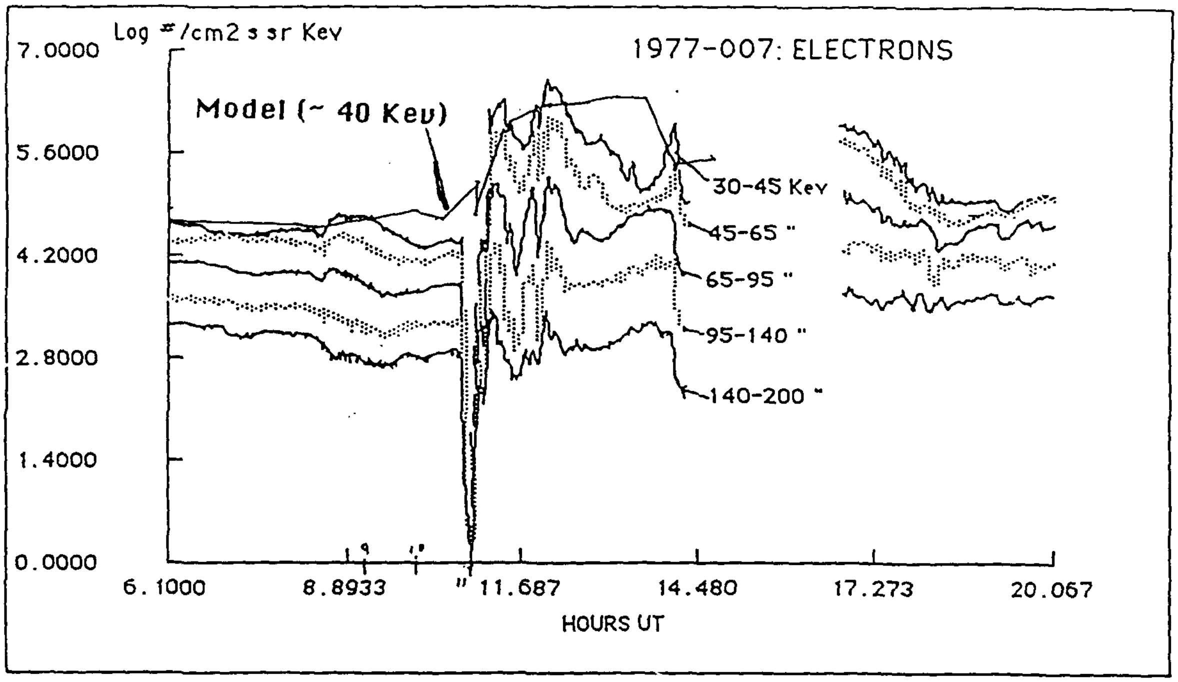
Previous runs have shown the 'Kivelson' effect is not important for 40 keV electrons at local midnight, and it is neglected for adjusting the η values. Solving equation 3.44 in Chapter 3, the revised density becomes

$$\eta_{tot}(r) = (1.966\text{E}21)\left[\frac{r}{15.87}\right]^{4.134}$$

The initial density values for entire grid are calculated using

$$\text{EETA}_{initial}(K,I,J) = \left[\frac{\eta_p\,(\text{midnight})}{1.966\text{E}21}\right]\left[\frac{r\,(I,J)}{15.87}\right]^{4.134}$$

where η_p(midnight) is the initial value for energy species K at midnight.



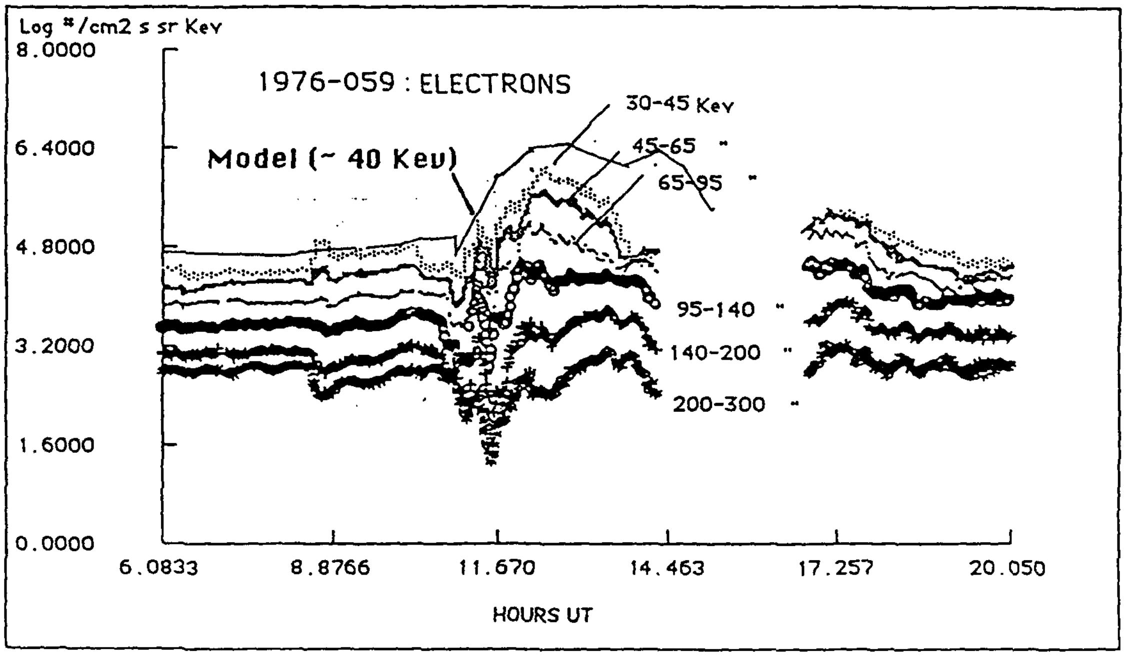


Figure 4.3 Calculated 40 keV geosynchronous electron flux values and 1977-077 and 1976-059 observed 40 keV electron flux values.

The plasma sheet depletion factor (f_{Fp}) defined in equation 3.35 of Chapter 3 was set to 5 and was activated at the substorm onset at 10:54:01 UT.

C. Model Results

Figure 4.4 shows the time evolution of the invariant density (η) in the equatorial plane for our highest energy electrons (~40 keV at synchronous orbit). Figure 4.5 is a similar plot for "medium energy" ions (~40 keV at synchronous orbit); and Figure 4.6 represents low energy ions (~2.5 keV at synchronous orbit). The contours represent contours of constant η; that is, constant numbers of particles per unit magnetic flux. The contours differ by a factor of $10^{0.2}$ in density level. In the plots, the sun is to the left. The outermost contour represents approximately the outer boundary of our calculations. Some physical features include

1) Collapse of the magnetic field causes the inner edge of the plasma sheet near local midnight to move earthward approximately 1 Re at the onset of the expansion phase of the first substorm at 1054UT.

2) Electrons drift eastward after the injection, with high densities reaching local noon at approximately 1200UT.

3) Medium energy ions drift westward from the midnight region, leading to high densities at local noon beginning about 1200UT.

4) Subsequently the afternoon inner edge moves gradually earthward.

The output from the model was compared with data available from the CDAW-6 interval, which included

1) Available data from geosynchronous orbit

a) 1-16 keV ions in the afternoon sector just after noon UT (GEOS-2)

b) 40 keV electrons around 0200 LT at 1100UT (1977-077)

c) 40 keV electrons around dawn at 1100UT (1976-056)

2) Low orbit data

$\lambda = 5750$ electrons (~ 40 keV at synchronous orbit)

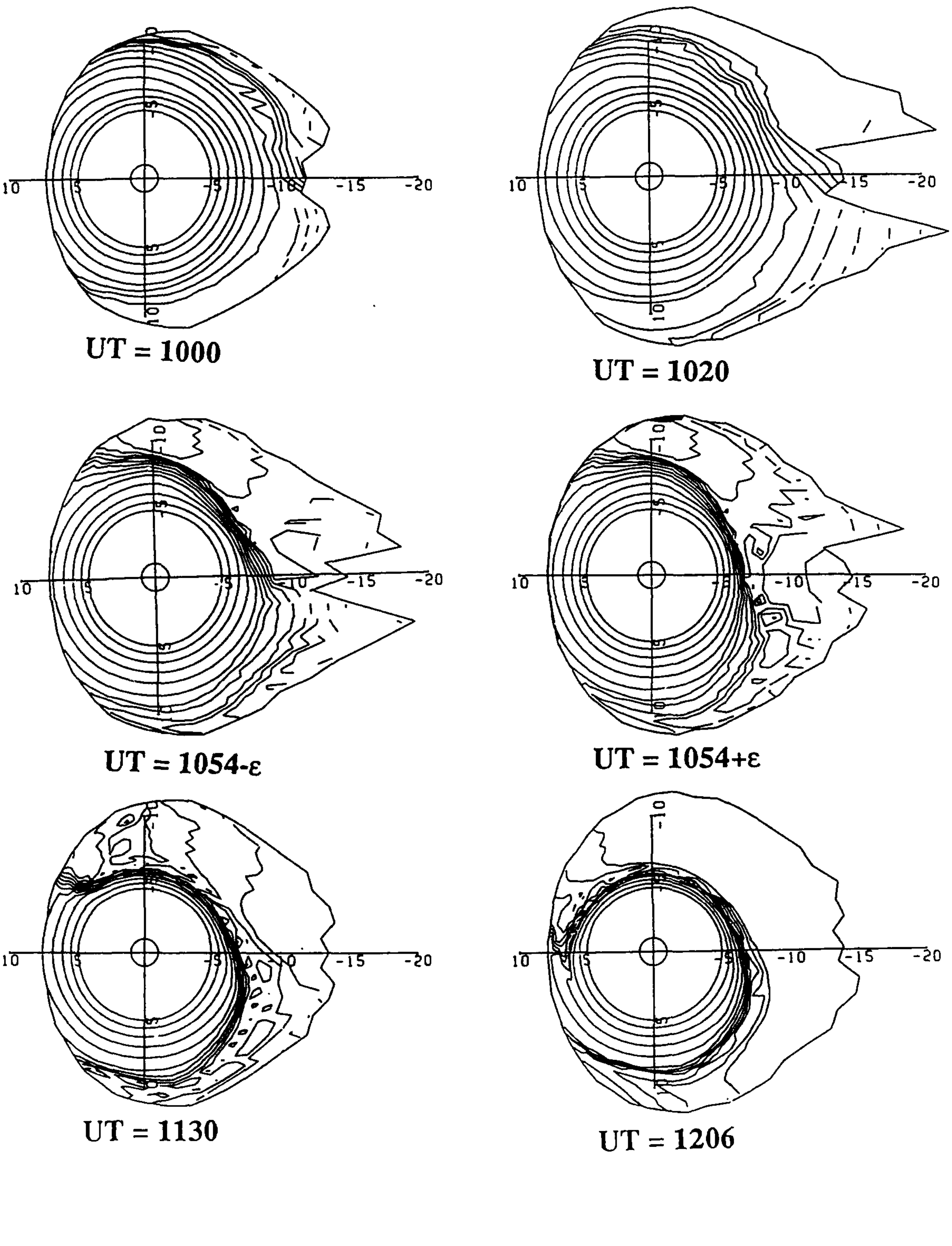


Figure 4.4 Time evolution of invariant density (η) in the uatorial plan for highest energy electrons (~40 keV at geosynchronous).

$\lambda = 5851$ ions (~ 40 keV at synchronous orbit)

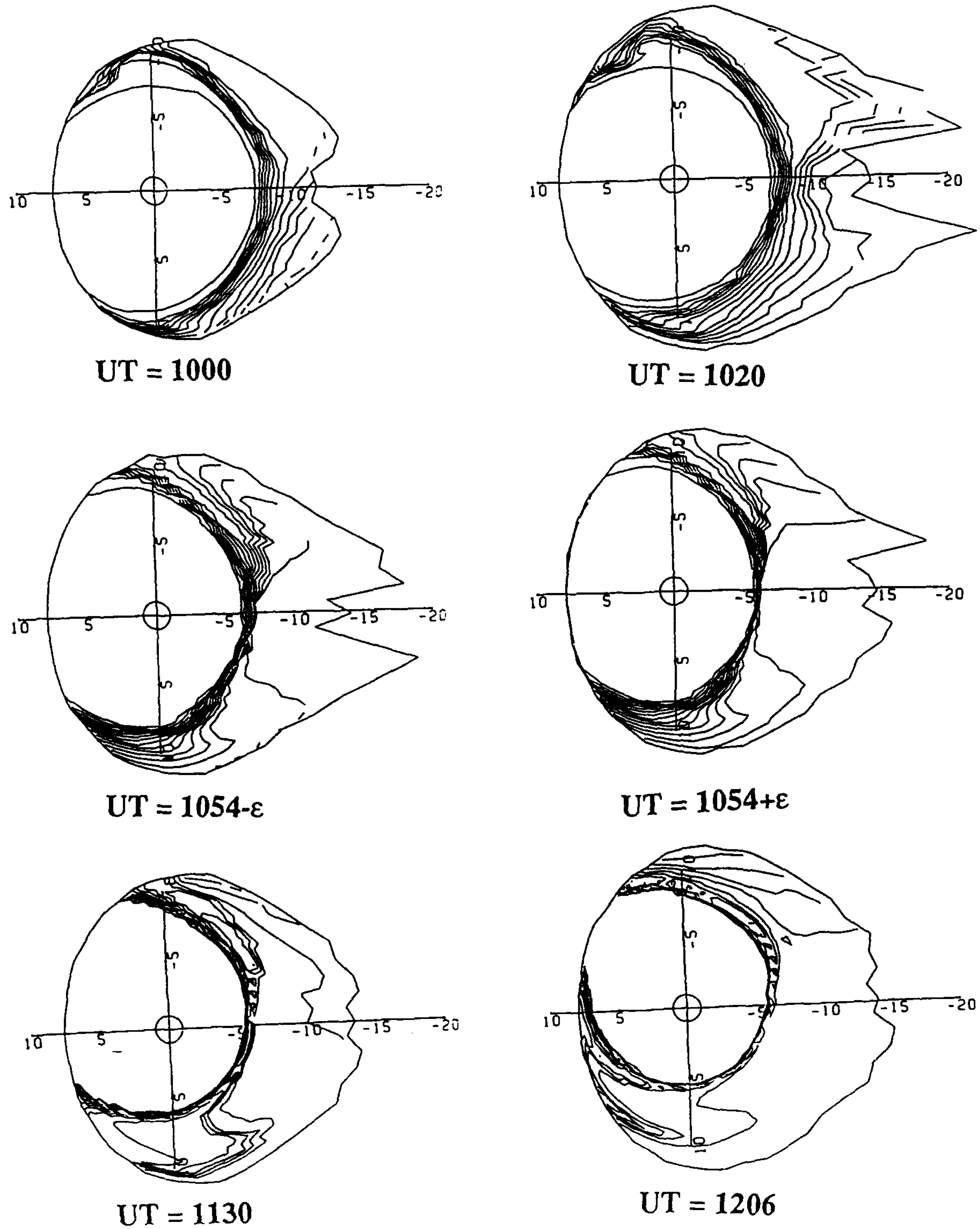


Figure 4.5 Time evolution of invariant density (η) in the equatorial plan for medium energy ions (~40 keV at geosynchronous).

λ = 375 ions (~ 2.5 keV at synchronous orbit)

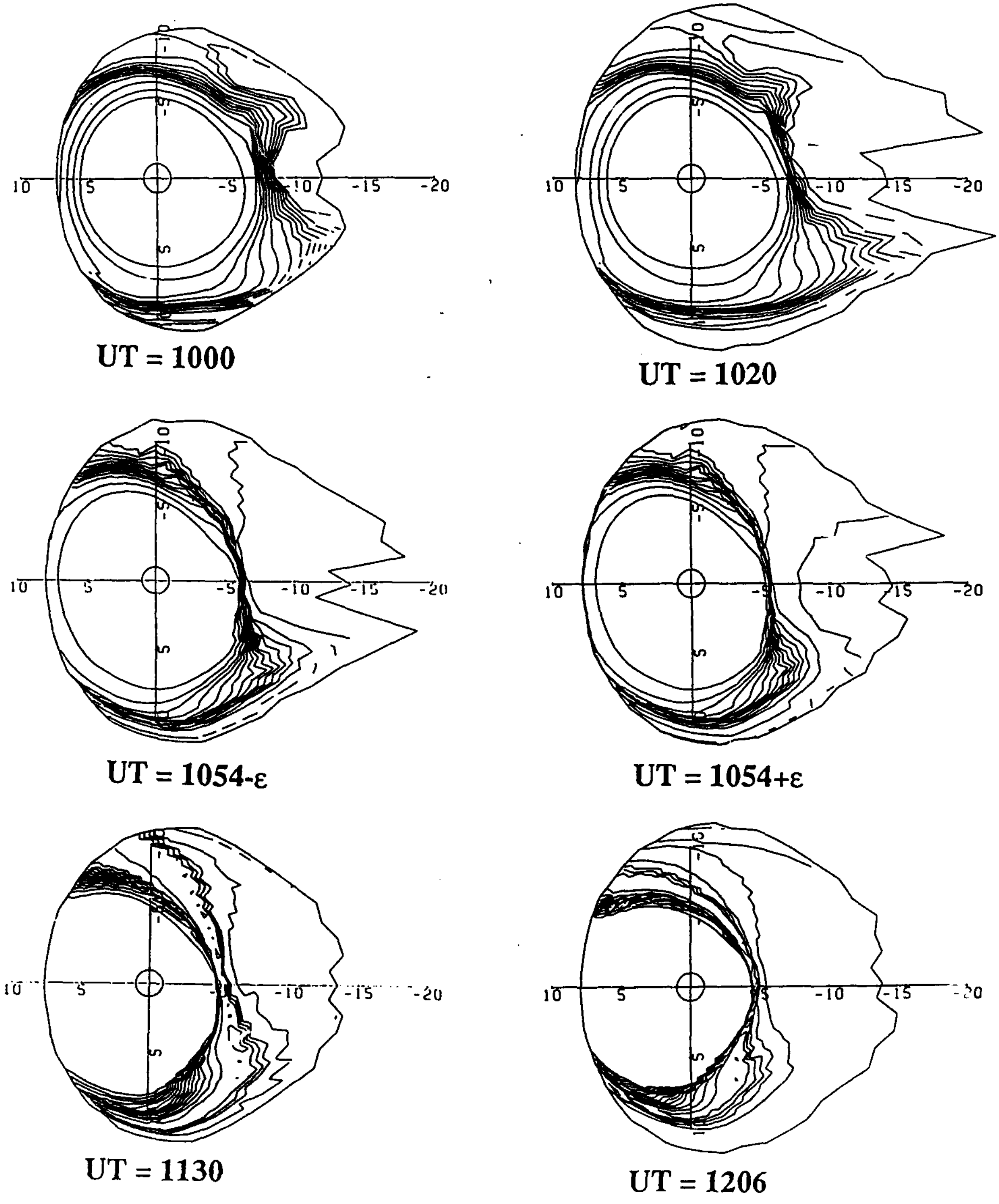


Figure 4.6 Time evolution of invariant density (η) in the equatorial plan for low energy ions (~2.5 keV at geosynchronous).

a) Precipitating electron flux from DMSP

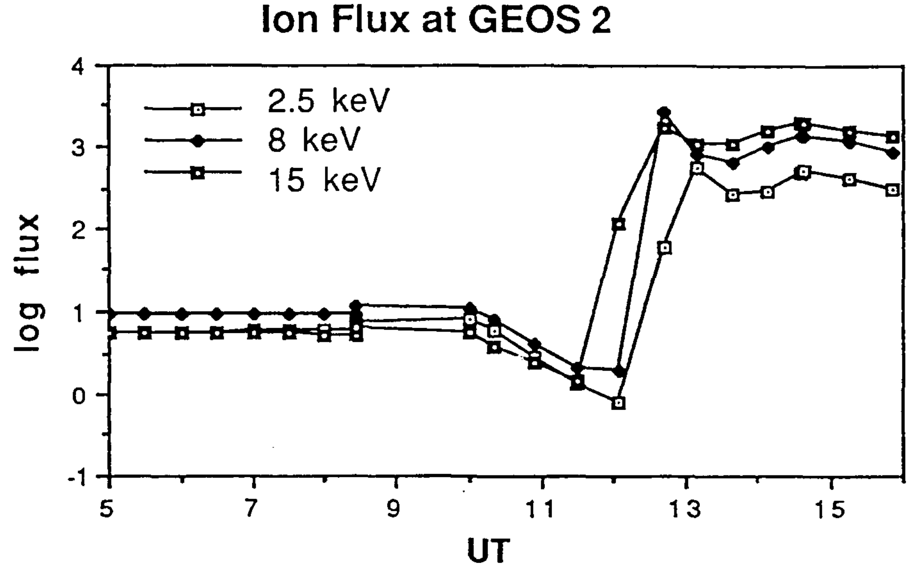
Figure 4.3 shows the electron flux measured at the geosynchronous satellites and compares them with the highest electron energy channel (K=32) in the model which corresponds to approximately 40 keV at geosynchronous orbit. The arrival times of approximately 1100UT for 1977-077 and 1130UT for 1976-059 show good agreement. However, in both cases the fluxes are too high by a factor of 2.

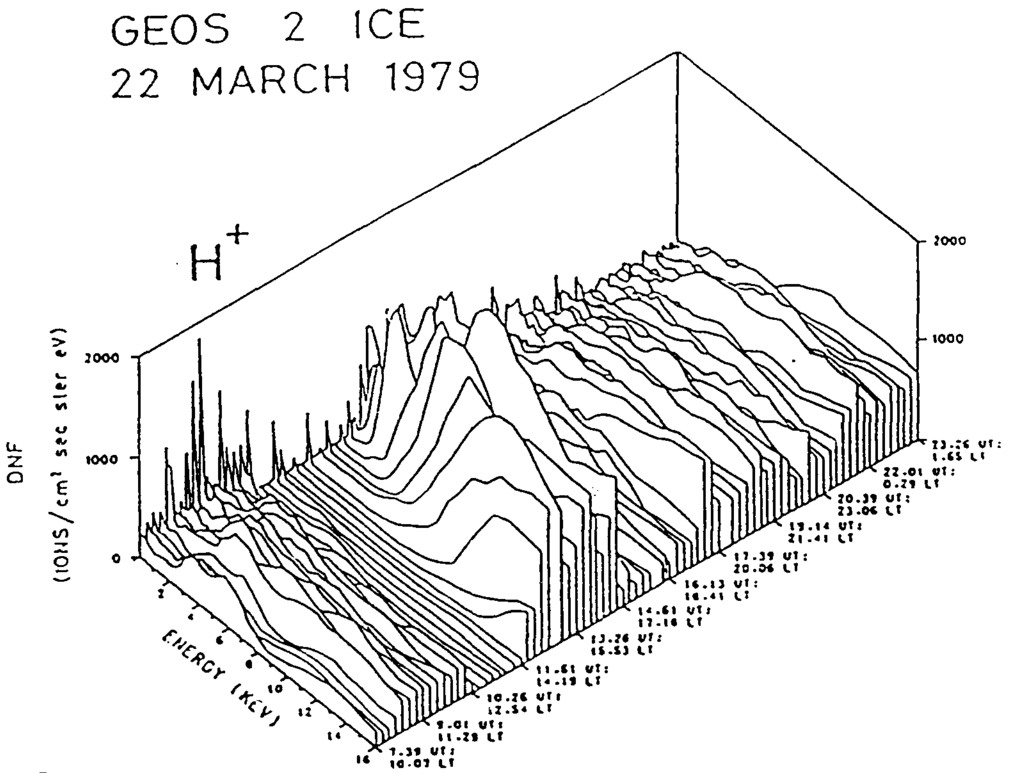
The ion energy channels for K=1, 2, and 3 in the model correspond approximately to geosynchronous energies of 2.5 keV, 8.0 keV, and 15.0 keV respectively. A comparison of the model ion flux with the GEOS-2 data is shown on Figure 4.7. Again the arrival times and energy dispersion are in good agreement with the 15 keV ions arriving around 1200UT, 8.0 keV ions arriving around 1230UT, and the 2.5 keV ions arriving about 1330UT. The peak fluxes show the same general tendency to be high by about a factor of 2.

The graphs of the DMSP precipitating electron energy flux are given as Figures 4.8 and 4.9. The model values normalized to the data are represented by overlays. The model does not calculate precipitation in the region poleward of our main modeling region; that excluded region includes the polar cap and the higher-latitude part of the auroral zone so this is not included. The data shown in Figure 4.8 are for the pre-substorm period. In all three examples both the shape and the magnitude of the fluxes agree well with the satellite data. Figure 4.9 presents the results after the substorm onset and Figure 4.10 is the substorm recovery period. As with the geosynchronous fluxes, for both cases the magnitude is again too high by a factor of 2.

D. Interpretation and Implication

This model represents an attempted simulation of the inner-magnetosphere processes associated with magnetospheric substorms, including the magnetic field collapse at the onset of the substorm expansion phase and the subsequent injection of particles into





After Stokholm, Amata, Balsiger, Candidi, Orsini, and Anderson, J. Geophys. Res., 90, pp 1253-1261, 1985

Figure 4.7 Calculated ion flux values and GOES 2 observed ion flux values.

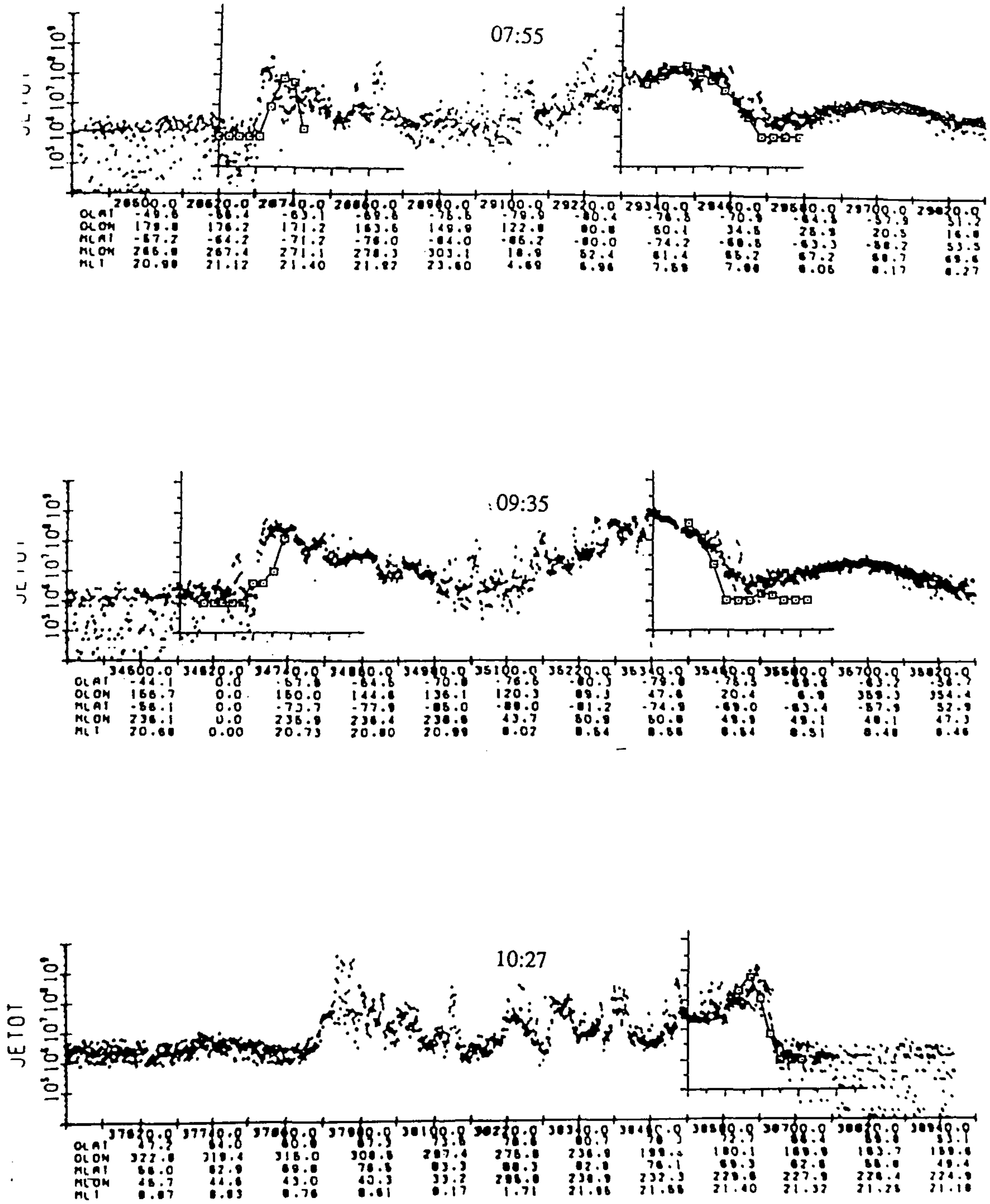


Figure 4.8 DMSP precipitating electron energy flux before substorm onset.

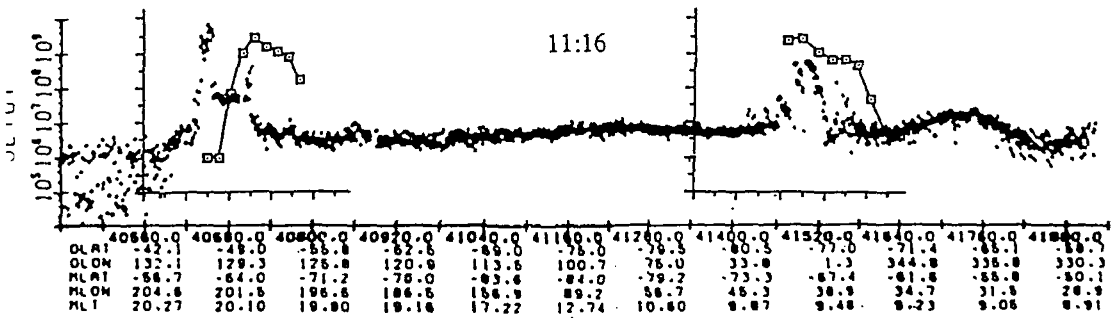
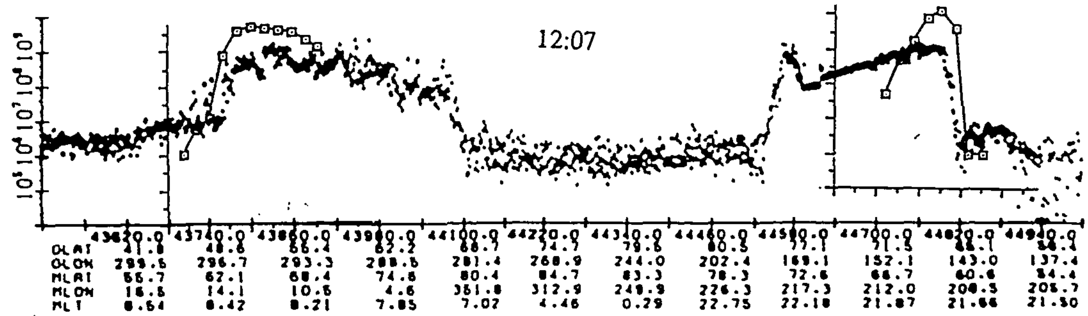


Figure 4.9 DMSP precipitating electron energy flux after substorm onset.



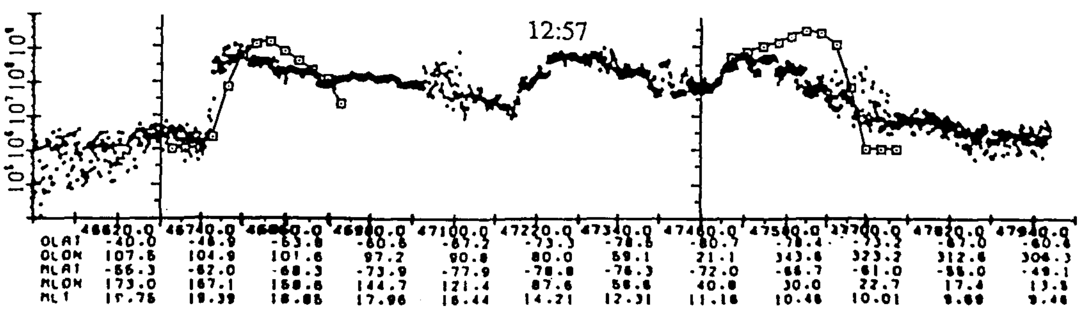


Figure 4.10 DMSP precipitating electron energy flux during substorm recovery period.

geosynchronous orbit. The arrival times of the electrons near dawn and the ions which have drifted to noon local time agree with the satellite data. However, assuming that middle plasma sheet flux tubes are convected adiabatically through the inner plasma sheet to synchronous orbit in a substorm with allowances for precipitation loss, the model results imply unrealistically high synchronous fluxes and unrealistically high precipitation rates. In addition, the modeled, geosynchronous 40 keV electron flux does not decrease during the recovery phase late in the event as the data shows. It may be that we are underestimating the pitch-angle scattering rate for these particles. The Hones (1979) substorm theory implies that a portion of the plasma on tail-plasma-sheet flux tubes escapes down the tail as a plasmoid. Therefore, for the initial model runs, we decreased the plasma sheet density by a factor of 2 at substorm onset. This still produced fluxes and precipitation too high by a factor of 5. The results presented here, from a later model run, assume a factor- of- 5 depletion after substorm onset. The model predictions and observations agree to within a factor of 2 in flux levels and precipitation. The implication is that the substorm process must involve a powerful electron loss process in addition to precipitation, or the flux tubes which convect into geosynchronous orbit have previously been depleted.

TEST ANALYSIS REPORT

Test Analysis Report

2. TEST ANALYSIS

2.1 Test The set of algorithms encoded as the MSM have been run and tested at Rice University. The results have been studied by comparing the model results with observed data. The version of the program code which was developed at Rice University gives the verified results. To validate the algorithms delivered to the Space Forecast Center, the output from the code run at Rice University will be compared with that run at the Space Forecast Center.

2.1.3 Performance. Using the output data furnished from Rice University, the output from the code run at the Space Forecast Center should be the same.

3. SUMMARY AND CONCLUSIONS

3.1 Demonstrated Capability. This test verifies that the algorithms developed for the Air Weather Service at Rice University are running properly at the Space Forecast Center.

DATABASE SPECIFICATION

Database Specification

2.1 Database Identification.

AWS Environmental Database - Presently all the input data are on disk files, some of which will no longer be needed when the Environmental Data Base is operational. Attached is a listing of the input parameters needed to run the MSM and a proposed data transfer method.

MSM Database - Below is a list of the input files needed to restart the model.

BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.
DKTABLE	Decay times for computing charge-exchange loss of ions
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model. (km/sec)
MSMIN	Input start and stop times, start direct-access record number, and sunspot number

Following is a list of files output from the MSM.

V	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

YMIN	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ZMIN	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BNDLOC	Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)
EFLUX	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)
FLXSUM	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
EAVG	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
AUGPAR	Output volume -number of data elements × number of time steps × 4 (bytes)
COLAT	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ALOCT	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
FLUX	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

2.3.1 Conceptual Model. The MSM will interrogate the Environmental Data Base at mark times for all physical parameters needed by the model. Attached is a copy of the MSM Control and Data Flow Diagram. Subroutine PARGEN within the MSM code will interpolate or extrapolate as necessary to adjust the data to the model time. Subroutine FECON then calculates the parameters needed within the program. The model is capable of running with no data except Kp. If Kp is missing, the program so states and stops.

Data from previous MSM runs is communicated through the direct-access record number read in file MSMIN. It specifies at which time label of the previous run to begin the current run. Zero means the start of a new run.

3.3.2 Content.

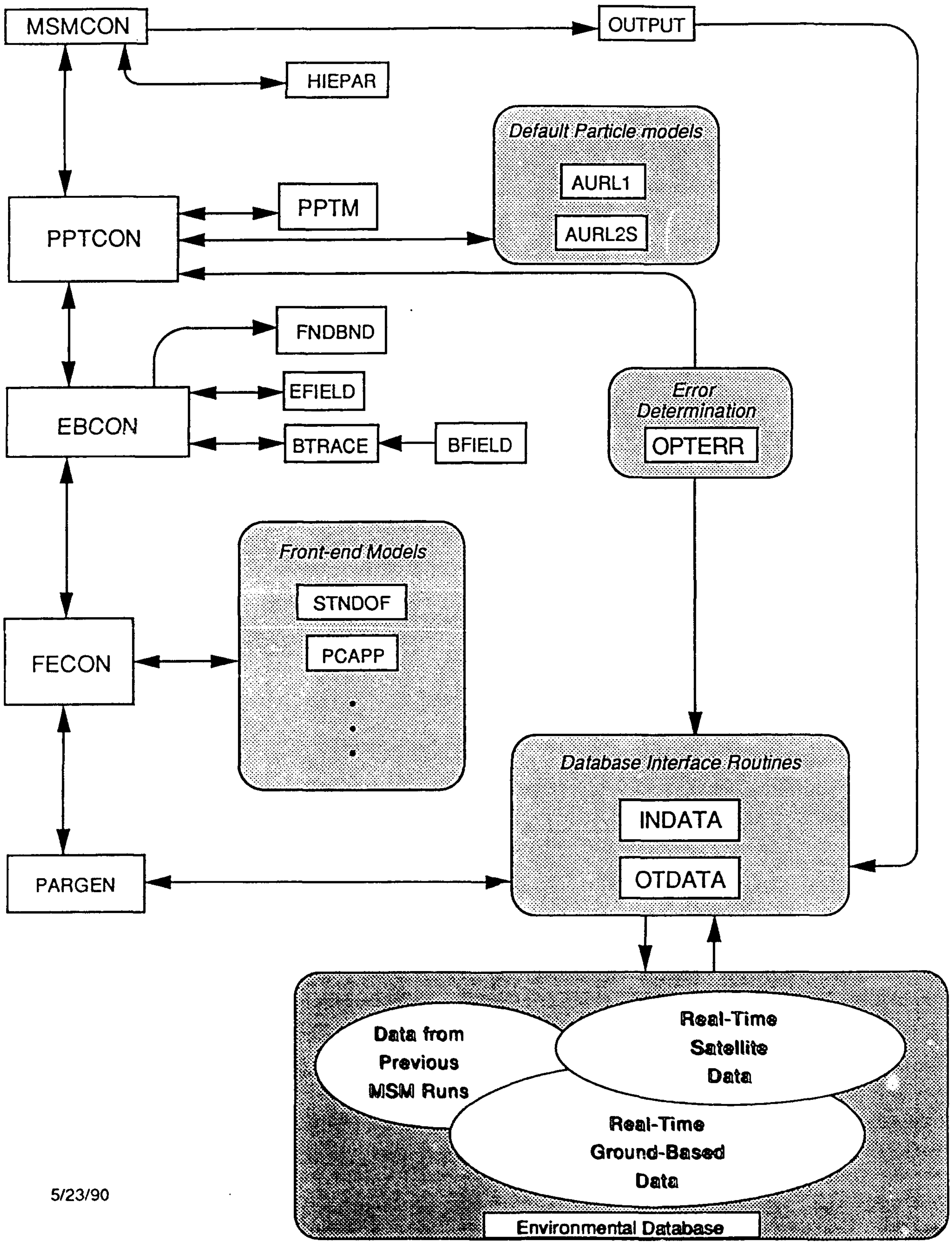
BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.

EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model. (km/sec)

Following is a list of files output from the MSM.

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume $((Re/nT)^{-2/3})$.
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm^2-sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) $O^{+)}$ (ergs/cm^2-sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOCT	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

MSM Control & Data Flow Diagram



5/23/90

PARAMETER	NAME	UNITS	I/O	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
INPUT ONLY:								
Kp	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(I)	EFLX(I)	$ELEC/CM^2$-S	I	X				X
GEO ION FLUX - CHANNEL(I)	GFLX(I)	$IONS/CM^2$-S	I	X				X
SOLAR WIND VELOCITY	SWVEL	KM/S	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	$PROT/CM^3$	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFBY	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

Data Transfer Method

1. MSM will access the environmental database input data files by calling a subroutine, **INDATA**.
2. The call will be of the form:
   CALL INDATA(parm-name, STARTT, ENDT, NDIM, DARRY, NUMNUM)
   Where:
   <u>parm-name</u> is a valid parameter name from the MSM input parameter list.
   <u>STARTT</u> is the starting time for this data request.
   <u>ENDT</u> is the ending time for this data request.
   <u>NDIM</u> is the horizontal dimension of DARRYand the maximum number of data points that may be returned.
   <u>DARRY</u> is a two dimensional array to be retrieved from the database. The form of DARRY is shown on the next page. The size of DARRY will be NDIM by 7.
   <u>NUMNUM</u> is the number of data values placed in DARRY by INDATA.

3. The system controller will periodically update the input data files.

4. Rice does not plan to write **INDATA**.

5. All parameters will be in single precision, floating point format.

6. Output data will be passed from MSM to the environmental database via a subroutine, **OTDATA**.

7. The call to **OTDATA** will be defined similarly to **INDATA**.

8. Rice does not plan to write OTDATA.

Format for DARRY

NDIM →

7 ↓

Data value at t1	Data value at t2	Data value at t3	Etc.
Time tag at t1	Time tag at t2	Time tag at t3	
Latitude	Latitude	Latitude	
Longitude	Longitude	Longitude	
Altitude	Altitude	Altitude	
mag. local time	mag. local time	mag. local time	

IMPLEMENTATION PROCEDURES

Implementation Procedures

3.1.5 Detailed Procedures. Following are the files needed to run the MSM, including those to be replaced by calls to the environmental data base. Those required for a restart are so labelled.

BOxxxxxx	The magnetic field matrices needed for the model run. Required for restart.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system. Required for restart.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
DKTABLE	Decay times for computing charge-exchange loss of ions
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model. Required for restart.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate. Required for restart.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete. Required for restart.
HARDY	Coefficients used by the Hardy precipitating electron model. Required for restart.
IONENG	Coefficients used as input for the ion precipitating model. Required for restart.
IONNUM	Coefficients used as input for the ion precipitating model. Required for restart.
MSMIN	Input start and stop times, start direct-access record number, and sunspot number
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm^3)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints $\times$ number of longitudinal gridpoints $\times$ number of time steps $\times$ 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts).

Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)

VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Equatorial magnetic field strength (nT). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces. Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)
EFLUX	Precipitating energy flux array ($ergs/cm^2$-sec). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) ($ergs/cm^2$-sec). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
AUGPAR	Augmented data array for input values. Output volume -number of data elements × number of time steps × 4 (bytes)
COLAT	Grid colatitude array (radians).

	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ALOCT	Grid local time array (radians eastward from noon).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
FLUX	Flux values for all energy channels at all grid points
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

4.1.2 Detailed Procedures. To run the MSM:

Start and stop times are specified to the program through the 5 line MSMIN file. Following is an example of MSMIN for the fresh startup of the program.

```
1988 112 64800              'start year, start day, start time (seconds)'
1988 112 66600              ' end year, end day, end time (seconds)'
0                           'direct-access record number 0 means new start'
'30 minute test run'        'up to 80 character run identification'
0.0                         'sunspot number=0.'
```

The MSMIN configuration given above would run the MSM for a 30 minute period from 1800 hours on day 112 of 1988 to 1830 on the same day. The sunspot number is zero.

To restart the MSM from 1830 (direct-access record number 3) and run until 2000 UT, the following MSMIN file would be used:

```
1988 112 66600
1988 112 72000
3
'MSM restart from 1830 on day 112'
0.0
```

Input file ENCHAN controls how many and which energy channels are calculated. The file set up is:

Number of energy channels to be modeled (maximum 30,integer)
Particle type 1 (integer), Energy 1 (floating point)
Particle type 2 (integer), Energy 2 (floating point)
. . . .
Particle type n (integer), Energy n (floating point)

The acceptable particle type input is:

1	electrons
2	H^+
3	O^+

The energy is at geosynchronous orbit in keV. The energies must be in ascending order, with particle types grouped together, which are checked within the program.

For example,

```
4
1 40
1 55
2 10
2 15
```

This is to model 4 energy channels, the first two are electrons of 40 and 55 keV at geosynchronous orbit. The second two are H^+ ions of 10 and 15 keV energy at geosynchronous orbit.

MAINTENANCE MANUAL

Maintenance Manual

2.1 System Application. The Magnetospheric Specification Model was developed as a set of algorithms to be incorporated into a real-time program for use in the U.S. Air Force Air Weather Service Space Forecast Center. It is intended to help Space Forecast Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.

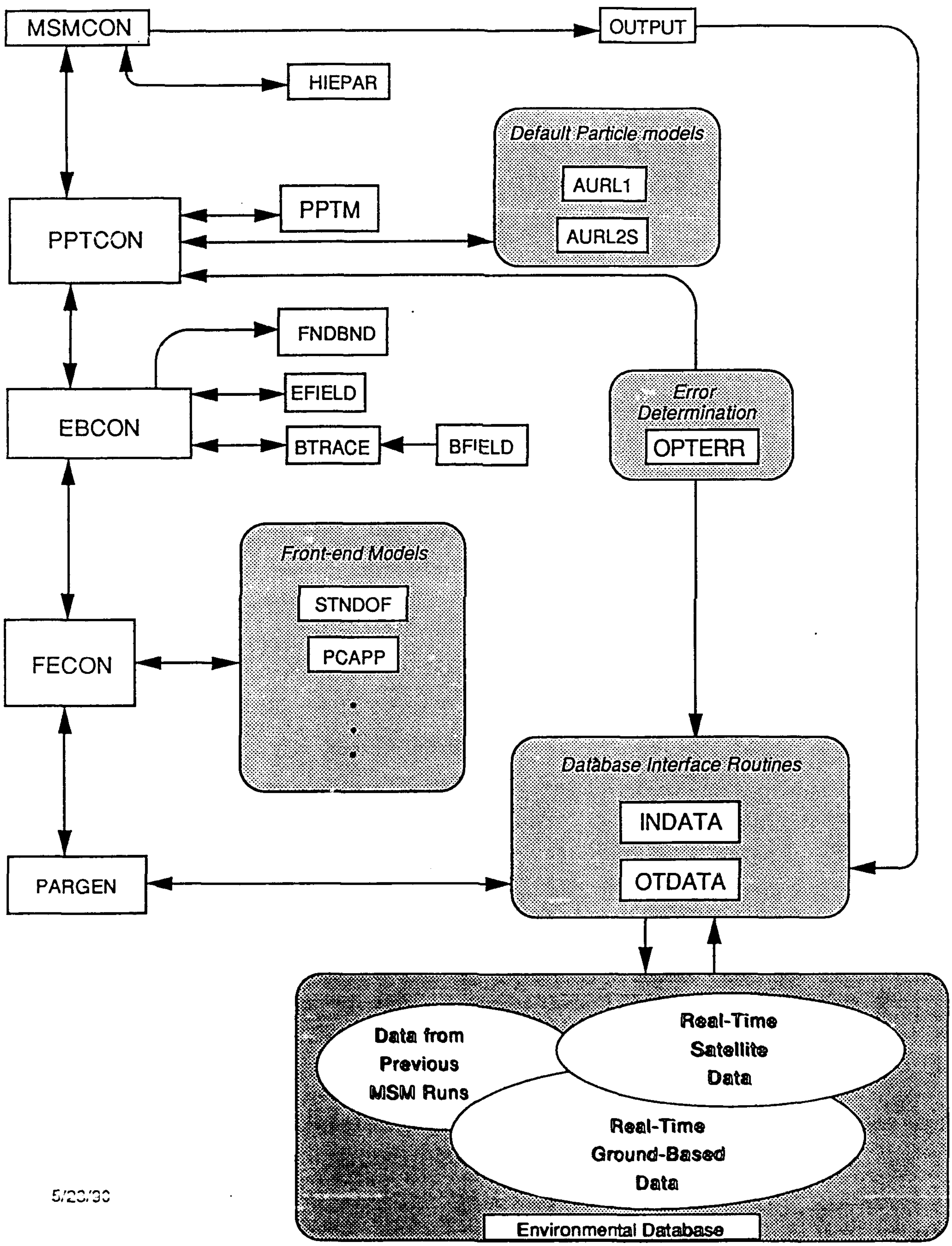
2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.2 System Organization. The model uses separate ionospheric electric-field and magnetospheric magnetic field models to trace particle trajectories in the magnetosphere equatorial plane. The following flowchart shows the major components of the model and how they are related.

4.1 Conventions. None.

5.1.1 Description. See following pages.

MSM Control & Data Flow Diagram



5/23/90

MSMCON

a. Functions - The MSM control program MSMCON provides dimensioning of the arrays used throughout the program, and the initial, end and increments of time information for all other subroutines of the program. It also accesses the high energy particle tracing routines.

b. Input

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day) January 1 is day 1.
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day) January 1 is day 1.
IEND(2)	End Time (seconds)
IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time step increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

c. Processing - following is a list of subroutines called and their major function:

PPTCON	Main control program for model
HIEPAR	High energy electron subroutine, currently a dummy routine pending computer code from sources outside Rice University.
OUTPUT	Output routine to interface with Space Forecast Center Environmental Database. Currently a dummy routine.

See the flowchart on the following page for the logic flow of the subroutine.

Global parameters for the run are set in this routine and are as follows:

LATDIM	Number of latitudinal grid lines (aliased as IDIM in some routines)
LTDIM	Number of local time (longitude) grid lines (aliased as JDIM in some routines)
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
KPPLUS	Kp dimension of the augmented FLXMAT array (KKPLUS = KPDIM + 2)

ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

d. Output. None. All output is currently handled by other subroutines.

PPTCON

a. *Functions* - main control program for the electric and magnetic field models and the running of the particle trace algorithm.

b. Input -

ISTART	Start year, day, and time (sec)
IINC	Major time step increment (year, day, time)
IEND	Ending year, day, and time (sec)
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
KPPLUS	Kp dimension of the augmented FLXMAT array (KKPLUS = KPDIM + 2)
ISPDIM	Plasma mass species dimension

c. Processing - following is a list of subroutines called and their major function:

RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities
OUTP	Utility subroutine to print out array information
FLXNIT	Subroutine to calculate Kp-based particle number flux in units of particles/cm^2/s/sr/eV as function of R, energy and Kp
DTIME	Routine which returns current system time
EBCON	Executive routine to control calculation of electric and magnetic field models
EPREAD	Dummy routine to read satellite flux data - to be replaced by call to environmental data base when program is operational
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program
BNDSET	Subroutine to compute boundary plasma distribution
RDHDR	Subroutine to read the header record of the standard MSM disk file format
INITAL	Subroutine to compute initial plasma distribution
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP

FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array - this routine is used as the MSM default model when full particle traces are not done
AURL1	Subroutine to determine default Hardy precipitating electron flux
AURL2S	Subroutine to determine default Hardy precipitating ion flux
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format
READ3D	Subroutine to read 3-d arrays from the standard MSM disk file format
RDVEC	Subroutine to read header vector from standard MSM disk file format
WRTVEC	Subroutine to write header vector from standard MSM disk file format
OPTERR	Subroutine to calculate standard deviation and error of the model flux and available observed geosynchronous satellite data

See the following flow-chart for the logical structure of the subroutine.

d. Output -

FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. ($\log_{10}$(#/cm^2-s-ster-eV))
CHID	Character header string (up to 80 characters)
ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV)

	RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour)
	RID(5) - RID(20) = Presently unused
SCTFRC	Assumed pitch angle scattering efficiency
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
TETA	Colatitude of I grid lines (radians)
PHI	Non-rotated hour angle of J grid lines (radians)
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
VNORTH	Northern hemisphere electric potential distribution (Volts)
VSOUTH	Southern hemisphere electric potential distribution (Volts)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere radius of ellipse measured in y (duskward) direction
DYS	Southern hemisphere radius of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere radius of ellipse measured in y (duskward) direction
DYN	Northern hemisphere radius of ellipse measured in y (duskward) direction
VM	Flux tube volume $((Re/nT)^{-2/3})$
VMLOSS	Geometric factor used in calculating precipitation loss
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ALAM	Energy invariant vector $(eV(Re/nT)^{2/3})$
ETABEG	Initial flux tube content distribution each time step $(weber^{-1})$
ETABND	Boundary flux tube content distribution $(weber^{-1})$
FLXBND	Flux at boundary $(cm^2\ second)^{-1}$
BNDLOC	Location of outer boundary of detailed particle traces
ETA	Flux tube content array $(weber^{-1})$

FLUX	Flux (cm^2 second)$^{-1}$
EFLUX	Precipitation energy flux array (ergs/cm^2-sec)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^{+}) (ergs/cm^2-sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
ALMDEL	Width of energy invariant channels
NAUGEL	Dimension of input arrays
AUGPAR	Augmented data array for input values
MODE	Logical variable telling which input parameters we have values for
RPP	Nominal radius of plasmapause defined as dipole mappine of equatorward edge of auroral zone to magnetospheric equatorial plane (Re)
THRSH	Vector giving full-traceback/default flux information
ERSHFT	Vector of error values
DKTIME	Array giving charge exchange decay times (seconds) for an array of radial distances, energies, sunspot numbers, and species

EBCON

a. Functions - EBCON calls routines to calculate the electric and magnetic field at our grid points for all time labels for this run.

b. Input -

ISTART	Start year, day, and time (sec)
IINC	Major time step increment (year, day, time)
IEND	Ending year, day, and time (sec)
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
LATDIM	Number of latitudinal grid lines (aliased as IDIM in some routines)
LTDIM	Number of local time (longitude) grid lines (aliased as JDIM in some routines)
ITMDIM	Maximum number of major time steps per run of the program
IEDIM	Number of invariant energy species traced by the program
NAUGEL	Number of elements in the augmented input array
CHID	Character header string (up to 80 characters)
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
IEMAX	Maximum number of energy channels
TETA	Colatitude of I grid lines (radians)

c. Processing - following are the subroutines called and their major function:

FECON	Subroutine to obtain time-normalized observational values
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid points
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency
BTRACE	Subroutine to calculate magnetic field models appropriate for geophysical conditions as specified by standoff distance, equatorward edge of the auroral zone, Dst, tail collapse, and tilt angle
EFBNDY	Subroutine to calculate latitudes and widths of electric field model boundarys 2 and 3
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V)
OUTP	Utility subroutine to print out array information
FNDBND	Subroutine to compute time dependent outer boundary location
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
VMLSET	Subroutine to calculate VMLOSS array
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

d. Outputs -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
VM	Flux tube volume (weber^{-1})
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
VNORTH	Northern hemisphere electric potential distribution (Volts)
VSOUTH	Southern hemisphere electric potential distribution (Volts)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere radius of ellipse measured in y (duskward) direction
DYS	Southern hemisphere radius of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere radius of ellipse measured in y (duskward) direction
DYN	Northern hemisphere radius of ellipse measured in y (duskward) direction
BNDLOC	Location of outer boundary of detailed particle traces
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ITMMAX	Maximum number of time labels

R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VMLOSS	Geometric factor used in calculating precipitation loss
SCTFRC	Assumed pitch angle scattering efficiency
AUGPAR	Augmented data array for input values
MODE	Logical variable telling which input parameters we have values for
RPP	Nominal radius of plasmapause defined as dipole mapping of equatorward edge of auroral zone to magnetospheric equatorial plane (Re)

BLOCK DATA

a. Functions - Block data containing the index of data elements and connections between the logical unit number and names. Sets I, J, and energy indices for locations and species to be traced.

b. Input - none.

c. Processing - No other subroutines are called.

The block data contains the data element assignments (for example for the AUGPAR vector) and connects the logical unit number to the name within the program.

d. Ouput - *the following assignments are made:*

Elements in the input data array

1	Year
2	Day
3	Seconds of day
4	Minute
5	Seconds
6	Kp
7	Dst (nT)
8	Equatorward edge of auroral zone at midnight (degrees)
9	Low latitude boundary of auroral zone (degrees)
10	Equatorward edge of Region 2 currents (degrees)
11	Currently unused
12	Bx component of interplanetary magnetic field (nT)
13	By component of interplanetary magnetic field (nT)
14	Bz component of interplanetary magnetic field (nT)
15	Magnetotail collapse parameter
16	X component of geosynchronous magnetic field data (nT)
17	Y component of geosynchronous magnetic field data (nT)
18	Z component of geosynchronous magnetic field data (nT)
19	Currently unused
20	Currently unused
21	Solar wind velocity (km/sec)
22	Solar wind density (prot/cm^3)
23	Tilt angle (degrees)
24	Standoff distance (Re)
25	Cross polar cap potential drop (kV)
26	Electric field pattern type
27	Time rate of change of Dst (nT/hour)

28	Time rate of change of the equatorward edge of the auroral oval at midnight (degrees/hour)

Elements in the logical unit assignment:

9	LUERR - Error output
13	LUPPT - Not currently used
14	LUCORD - Coordinate data
15	LUPRNT - Output to printer
16	LUIDAT - Not currently used
17	LUHDYE - Coefficients to calculate electron precipitating energy and number flux in default model
20	LUDK - charge exchange decay parameter file
21	LUETAB - ETABND file (boundary flux tube content distribution)
22	LUEBEG - ETABEG file (initial flux tube content distribution each time step)
23	LUSHFT - error correction factor file
24	LUPION - Not currently used
25	LUEAVG - EAVG file (average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+)
26	LUFLSM - FLXSUM file (total precipitation energy flux array (integrated over species for (1) electrons, (2) H^+ and (3) O^+)
27	LUFLX - FLUX file (particle flux array)
28	LUEFLX - EFLUX file (precipitation energy flux array)
35	LUEFLD - V file (electric potential distribution on grid (average of VNORTH and VSOUTH)
36	LUBFLD - VM file (flux tube volume)
37	LUBMIN - BMIN file (equatorial magnetic field strength (nT))
38	LUXMIN - XMIN file (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
39	LUYMIN -YMIN file (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
40	LUZMIN -ZMIN file (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
41	LUALOC - ALOCT file (grid local time array (radians eastward from noon))
42	LUCOLT - COLAT file (grid colatitude array (radians))
43	LUBNDL - BNDLOC file (location of outer boundary of detailed particle traces)
44	LUPDAT - AUGPAR vector (augmented data array for input values)
45	LUVN - VNORTH file (northern hemisphere electric potential distribution)
46	LUVS - VSOUTH file (southern hemisphere electric potential distribution)
47	LIONEG - Coefficients to calculate ion precipitating energy flux in default model
48	LIONNO - Coefficients to calulate ion number flux in default model
49	LUENCH - File used to input number and range of energy channels to model

Other assignments made:

IBEG	Starting I value, currently set to 1
ISTOP	Ending I value, currently set to 53
JBEG	Starting J value, currently set to 3
JSTOP	Ending J value, currently set to 50
IEBEG	Starting energy value index, currently set to 1

EMODEL

a. Functions - This subroutine specifies the electric potential at each grid point in the modelling region for the northern hemisphere, southern hemisphere and the average of both hemispheres.

b. Inputs

IPATT	Polar cap pattern type (from Heppner-Maynard empirical model)
PCP	Cross polar cap potential (Volts)
DEQDT	Time rate of change of the latitude of the equatorward edge of the auroral zone at midnight (degrees/hour)
A	Radius of ellipse measured in X(sunward) direction
B	Radius of ellipse measured in Y(duskward) direction
DX	Sunward displacement of coordinate system center from pole
DY	Duskward displacement of coordinate system center from pole
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
MODE	Not currently used
LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
JWRAP	Number of points of overlap in J (longitude) direction
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes , but scaled to externally specified cross polar cap potential drop and ellipse parameters = -1 means Heppner-Maynard formula is used for all latitudes, unscaled. = -2 should not be used. It is the form used by the Rice Convection Model
ITOP	= 1 means the ellipse parameters A, B, DX, and DY for the polar cap boundary are computed internally in the subroutine. This is the normal mode for the MSM.

c. Processing - following is a list of the subroutines called and their major function:

INPUT	Get input needed for computation of Heppner-Maynard model
THET	Function that gives equation for the electric field boundary ellipse in flat polar coordinates
EPOT	Using scaled Heppner-Maynard model returns potential at a specific point
LOW	Returns potential at a specific location in low latitude region
AURORA	Returns potential at a specific location in auroral region
REG1	Returns potential at a specific location in Region 1

A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

V	Electric potential disbritution on grid (Volts)

EPOT

a. Functions - This subroutine uses the scaled Heppner-Maynard model to return the electric potential at a specific point.

b. Input -

COEF	Coefficients from digitization of Heppner-Maynard electric field model
TLAT	Colatitude of point on boundary to be calculated (radians)
TLON	Magnetic local time of point on boundary to be calculated (radians)
NTAPE	Unit number from which to read Heppner-Maynard coefficients
IPATT	Heppner-Maynard pattern type
NNMAX	Number of coefficients to use in calculation
A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Radius of ellipse measured in y (duskward) direction
DY	Radius of ellipse measured in y (duskward) direction
XCO	Electric field model coefficients
AHM	Radius of ellipse measured in x (sunward) direction from Heppner-Maynard Model
BHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DXHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DYHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
VMIN	Computed electric potential minimum from Heppner-Maynard Model
VMAX	Computed electric potential maximum from Heppner-Maynard Model
PCP	Cross-polar-cap potential drop
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes , but scaled to externally specified cross polar cap potential drop and ellipse parameters = -1 means Heppner-Maynard formula is used for all latitudes, unscaled. = -2 should not be used. It is the form used by the Rice Convection Model

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VALUE	Electric potential at point on boundary a (Volts)

LOW

a. Functions - This subroutine returns the electric potential at a specification location in the low latitude region of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
ALO	Magnetic local time of point on boundary to be calculated (radians)
DEQDT	Time rate of change of equatorward edge of auroral zone (degrees/hour)
AEQEDG	Radius of ellipse 3 measured in x (sunward) direction
DXEQEG	Offset of ellipse 3 in x direction
VBAR	Average potential at low latitudes
G	Weighting factor

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VLOW	Electric potential of a point in low latitude region (Volts)

AURORA

a. Functions - This subroutine returns the electric potential at a specific location in the auroral zone region of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
ALO	Magnetic local time of point on boundary to be calculated (radians)
VBAZ	Auroral-zone contribution to the potential at boundary b
TB	Location of boundary b for specific local time
TC	Location of boundary c for specific local time
J	Local time index
JMAX	Maximum number of points in J direction
JWRAP	Number of points of overlap in J direction

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VAZ	Electric potential for a point in the auroral zone (Volts)

REG1

a. Functions - This subroutine returns the electric potential for a specific point in Region 1 of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
AL	Magnetic local time of point on boundary to be calculated (radians)
TA	Location of boundary a for specific local time
TB	Location of boundary b for specific local time
VAA	Electric potential for point on boundary a
VBB	Electric potential for point on boundary b
DVAA	Derivative of electric potential for point on boundary a
DVBB	Derivative of electric potential for point on boundary b

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

V1	Electric potential for a point in Region 1 (Volts)

THET

a. Function - This function subprogram gives the equation for an electric field boundary ellipse in flat polar coordinates.

b. Input -

AA	Radius of ellipse measured in X(sunward) direction
BB	Radius of ellipse measured in Y(duskward) direction
XC	Sunward displacement of coordinate system center from pole
YC	Duskward displacement of coordinate system center from pole
PHI	Magnetic local time

c. Processing - This function gives the solution to the equation for an ellipse in flat polar coordinates. It specifies colatitude as a function of local time.

d. Output -

THET	Colatitude of input ellipse parameters and local time

INPUT

a. Function - This subroutine reads in the necessary input for the computation of the Heppner-Maynard electric field model.

b. Input -

NTAPE	Unit number from which to read Heppner-Maynard coefficients

c. Processing - This subroutine reads from logical unit NTAPE the coefficients necessary for the Heppner-Maynard electric field model.

d. Output -

COEF	Coefficients from digitization of Heppner-Maynard electric field model
IPATT	Polar cap pattern type (from Heppner-Maynard empirical model)
XCO	E field coefficients
AHM	Radius of ellipse measured in x (sunward) direction from Heppner-Maynard Model
BHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DXHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DYHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
VMIN	Computed electric potential minimum from Heppner-Maynard Model
VMAX	Computed electric potential maximum from Heppner-Maynard Model

BTRACE

a. Functions - Subroutine to calculate magnetic field models for appropriate geophysical conditions as specified by the standoff distance, equatorward edge of the auroral zone, Dst, tail collapse parameter, and tilt angle. Calculates the output on the MSM spatial grid by interpolating between pre-computed magnetic field models.

b. Input -

LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program
L	Current temporal grid index
FSTOFF	Standoff distance (Re)
FEQEDG	Equatorward edge of aurora (degrees)
FDST	Dst (nT)
FCLPSE	Collapse parameter
FTILT	Tilt angle (degrees)
ALOCT	Local time hour angle (measured east from noon)
TETA	Colatitude of I grid lines (radians)

c. Processing - following are the subroutines called and their major functions:

GETMAT	Subroutine to find the magnetic field matrices needed for interpolation in order to represent current conditions
RMVBSH	Subroutine to extrapolate magnetic field matrices into extremely stretched or open regions
BFGYRO	Subroutine to rotate fixed grid magnetic field arrays to match rotating corrdinate system

This subroutine determines which of the pre-computed magnetic field matrices are needed to match the geophysical conditions specified . After the matrices are retrieved and stored into a work array (WORK(LATDIM,LTDIM,2,2,2,2,2) they are interpolated using a multilinear interpolation scheme given by Press et al. in Numerical Recipes, Cambridge University Press, 1987, pp 95-97. The arrays are then rotated so that they have the same local time orientation as the main program.

d. Output -

VM	Flux tube volume $((Re/nT)^{-2/3})$
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)

ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane

BFGYRO

a. Functions - Subroutine to rotate fixed grid magnetic field arrays to match the rotating grid system of the main program.

b. Inputs -

LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
ARRAY	Unrotated magnetic field values called sequentially for the following quantities: VM - Flux tube volume $((Re/nT)^{-2/3})$ BMIN - Equatorial magnetic field strength (nT) XMIN - GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) YMIN - GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) ZMIN - GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
WORK	Internal working array
ALOCT	Local time hour angle (measured east from noon)

c. Processing - No other subroutines are called.

This routine uses a linear interpolation scheme to determine how much the magnetic field matrices should be rotated (F). It applies this factor to the input matrices so that the magnetic local time of the magnetic field matrices and the main program agree.

d. Output -

ARRAY	Rotated magnetic field values for the following quantities: VM - Flux tube volume $((Re/nT)^{-2/3})$ BMIN - Equatorial magnetic field strength (nT) XMIN - GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) YMIN - GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) ZMIN - GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)

PPTM

a. Functions - This subroutine traces particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP.

b. Inputs

TSTART	Time at which to begin particle traceback (seconds)
TSTOP	Time at which to end particle traceback (seconds)
DTNOM	Nominal time step to begin traces. DTNOM < 0 to trace back in time. PPTM will adjust time step as needed
SCTFRC	Assumed pitch angle scattering efficiency
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
ITMMAX	Number of temporal grid points used
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
KPPLUS	KP dimension of augmented FLXMAT array (KPPLUS=KPDIM+2)
ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. ($\log_{10}$(#/cm^2-s-ster-eV))
TIMTAG	Vector giving times at which E and B parameters are calculated
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$)
ALMDEL	Width of energy invariant channels
VM	Flux tube volume ((Re/nT)$^{-2/3}$)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane

VMLOSS	Geometric factor used in calculating precipitation loss
V	Electric potential distribution on grid (Volts)
ETABEG	Initial flux tube content distribution each time step ($weber^{-1}$)
ETABND	Boundary flux tube content distribution ($weber^{-1}$)
BNDLOC	Location of outer boundary of detailed particle traces
LL	Current temporal grid index
RPP	Nominal radius of plasmapause defined as dipole mapping of equatorward edge of auroral zone to magnetospheric equatorial plane (Re)
FKP	Current Kp value
IEMAX	Number of energy channels in use
THRSH	Vector giving full-traceback/default flux information
TSHENG	Threshold value to switch from full-traceback to default models - current set at > 100 keV electrons
TSHING	Threshold value to switch to default models for ions - currently set at 50 keV
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array
DKTIME	Array giving charge exchange decay times (seconds) for an array of radial distances, energies, sunspot numbers, and species

c. Processing - following are the subroutines called and their major function

SETREF	Subroutine to set upper limit reference flux
OUTP	Utility routine to print our array information
TNORML	Utility function subprogram to calculate run-normalized time
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
WKRATE	Function subprogram to evaluate weak precipitation loss rate
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction
MODFIX	Subroutine to adjust BBJ to be between 2 and JDIM-1
TCHK	Subroutine to check whether time boundary was crossed
BNDCHK	Subroutine to determine whether particle has crossed boundary of calculation
FLXVAL	Function subprogram to calculate Kp-dependent flux value a R=RR for particles of energy ENRG and Kp=FKP by interpolating FLXMAT array

See the following flow chart for the logic of the subroutine.

d. Output

ETA	Flux tube content array ($weber^{-1}$)

EFLUX	Precipitation energy flux array (ergs/cm^2-sec)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) (ergs/cm^2-sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV)

BNDCHK

a. Functions - This subroutine determines whether a particle has crossed the spatial boundary of the model calculation.

b. Input -

T	Time
IE	Energy channel
BBI	I location at end of time step
BBJ	J location at end of time step
BIOLD	I location at beginning of time step
BJOLD	J location at beginning of time step
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ETABND	Boundary flux tube content distribution ($weber^{-1}$)
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
BNDLOC	Location of outer boundary of detailed particle traces
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
ITMMAX	Number of temporal grid points used
JWRAP	Number of points of overlap in J direction

c. Processing - following are the subroutines called and their major function:

TNORML	Utility function subprogram to calculate run-normalized time

Using BNDMAX, the I value of the boundary, the time-normalized I (BBI) and J (BBJ) input values are compared to see if the particle has passed out of the outer boundary of the calculation region. If the particle has passed the boundary, the ETA value at that point on the boundary is calculated and returned.

d. Output -

ETANIL	If boundary is crossed, ETANIL is the value of ETA at the boundary crossing point ($weber^{-1}$)
IEXIT	If the boundary is crossed, IEXIT = 2

DVEFDI

a. Function - Function subprogram to compute the I derivative of the effective potential at grid point (I,J) at normalized time BT.

b. Input -

I	I coordinate of the grid point
J	J coordinate of the grid point
BT	Normalized time
AALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ((Re/nT)$^{-2/3}$)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

Standard central-difference formula is used to differentiate the effective potential (potential energy - kinetic energy) in the I direction.

d. Output -

DVEFDI	Derivative of effective potential at point (I,J,BT)

DVEFDJ

a. Function - Function subprogram to compute the J derivative of the effective potential at grid point (I,J) at normalized time BT.

b. Input -

I	I coordinate of the grid point
J	J coordinate of the grid point
BT	Normalized time
AALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ((Re/nT)$^{-2/3}$)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

Standard central-difference formula is used to differentiate the effective potential (potential energy - kinetic energy) in the J direction.

d. Output -

DVEFDJ	Derivative of effective potential at point (I,J,BT)

G3NTRP

a. Functions - Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

b. Input -

A	Array to be interpolated
IMAX	I dimension of array A
JMAX	J dimension of array A
KMAX	K dimension of array A
BI	Floating point value to interpolate in I dimension
BJ	Floating point value to interpolate in J dimension
BK	Floating point value to interpolate in K dimension

c. Processing - No other subroutines are called.

This subroutine uses the standard linear interpolation technique to interpolate an array in 3 dimensions.

d. Output -

G3NTRP	Interpolated point from array A

MODFIX

a. Functions - Subroutine to adjust J value to be within array bounds.

b. Input -

BBJ	J value to be adjusted
JDIM	J dimension of array
JWRAP	Number of points of overlap in J direction

c. Processing - No other subroutines are called.

The subroutine tests to see if the value is within the array bounds and adjusts the value if it is not.

d. Output -

BBJ	Adjusted J value

MOVER

a. Functions - Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction.

b. Input -

T	Time parameter
DTTRY	Time step to try
	< 0 = moves particles back in time
	> 0 = moves particles forward in time
BIOLD	Current I location of particle
BJOLD	Current J location of particle
EPS	Accuracy parameter
BBISCL	I error scaling parameter
BBJSCL	J error scaling parameter
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ($(Re/nT)^{-2/3}$)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major functions:

RK4	4th order Runge-Kutta routine

This routine is based on the discussion in Numerical Recipes, Press et al, on pages 550-560 in the chapter on integration of ordinary differential equations.

d. Output -

BBI	Updated I location of particle
BBJ	Updated J location of particle
DTUSED	Time step actually used
DTNEXT	Estimated next stepsize

RK4

a. Functions - This subroutine integrates an ordinary differential equation using a 4th order Runge-Kutta scheme.

b. Inputs -

T	Time parameter
DT	Time step
	< 0 = moves particles back in time
	> 0 = moves particles forward in time
BBIIN	Current I location of particle
BBJIN	Current J location of particle
AALAM	Energy invariant vector $(eV(Re/nT)^{2/3})$
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume $((Re/nT)^{-2/3})$
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major function:

VLOCTY	Subroutine to compute I and J components of particle velocity with energy invariant AALAM at location (BI,BJ) at normalized time BT

This algorithm comes from Abramowitz, Milton and Stegun, Irene A., 1964, Handbook of Mathematical Function, National Bureau of Standards, 1970, p. 896, section 25.5.10. It is also referenced in Numerical Recipes by Press et al.

d. Output

BBIOUT	Computed I location at end of time step
BBJOUT	Computed J location at end of time step

TCHK

a. Functions - This subroutine checks whether a particle has crossed a time boundary for this time step.

b. Input -

T	Time
TSTOP	Time at which to end particle traceback
DTNOM	Nominal time step to begin traces
BBI	I location of particle
BBJ	J location of particle
IEUSE	Energy index of particle
ETABEG	Initial flux tube content distribution each time step ($weber^{-1}$)
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program

c. Processing - following are the subroutines called and major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

If time boundary is crossed, the ETA at the location (BBI,BBJ) is interpolated.

d. Output -

ETANIL	ETABEG value at location (BBI,BBJ) at time T for energy species IEUSE if particle has crossed time boundary ($weber^{-1}$)
IEXIT	= 1 if particle has crossed time boundary

TNORML

a. Functions - Utility function subprogram to calculate run-normalized time.

b. Input -

TT	Time (seconds)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ITMMAX	Number of temporal grid points used

c. Processing - No other subroutines are called.

Using the run times given in TIMTAG, the time is located. Using standard linear interpolation, the correction factor is calculated.

d. Output -

TNORML	Non-integer time index corresponding to time TT

VLOCTY

a. Functions - This subroutine computes the I and J components of particle velocity for a particle with energy invariant AALAM at location (BBI,BBJ) at normalized time BT.

b. Input -

BBII	I location of particle
BBJJ	J location of particle
BT	Normalized time
AALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$)
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ((Re/nT)$^{-2/3}$)
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

This subroutine computes the I and J components of the particle's ExB drift.

d. Output -

DBIDT	I velocity of particle (I units/sec)
DBJDT	J velocity of particle (J units/sec)

RDGRID

a. Functions - Subroutine to read grid system coordinates and calculate essential grid quantities.

b. Input - Read from input file 'COORD'

ID	Eight character identifier
OFFSET	Offset of center of coordinate system from magnetic pole, currently zero
DLAM	Latitudinal grid spacing constant
DPSI	Local time grid spacing constant
RI	Radius of earth + ionosphere (km)
RE	Radius of earth (km)
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
TETA	Colatitude of I grid lines (radians)
PHI	Non-rotated hour angle of J grid lines (radians)

c. Processing - No other subroutine are called.

d. Additional outputs -

BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
COLAT	Grid colatitude array (radians)

MLTSET

a. Functions - This subroutine calculates the time dependent magnetic local time of rotating grid points.

b. Input -

ISEC	Time from start of run (seconds)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines

c. Processing - No other subroutines are called.

The subroutine calculates the change in the hour angle of the J grid lines (DPHI) and the time rate of change of the hour angle of the J grid lines (DPHIDT) to compute the time dependent magnetic local time grid

d. Output -

ALOCT	Time dependent magnetic local time grid

VMLSET

a. Functions - This subroutine sets up the time-dependent VMLOSS array

b. Input -

ITMCUR	Current time label
VM	Flux tube volume $((Re/nT)^{-2/3})$
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

d. Ouput -

VMLOSS	Geometric factor used in calculating precipitation loss

FNDBND

a. Functions - This subroutine computes the time dependent outer model boundary location.

b. Input -

ITMCUR	Current time label
COLAT	Grid colatitude array (radians)
ALOCT	Time dependent magnetic local time grid (radians eastward from noon)
A2	Radius of ellipse 2 measured in x (sunward) direction
B2	Radius of ellipse 2 measured in y (duskward) direction
DX2	Offset of ellipse 2 measured in y (duskward) direction
DY2	Offset of ellipse 2 measured in y (duskward) direction
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major function:

THET	Function that gives the equation for an electric field boundary ellipse in flat polar coordinates

For each J (longitudinal) value, the location of the outer boundary is calculated. Then the colatitude array (COLAT) is searched to find the pair of I values that encompass the boundary. The location is then linearly interpolated and placed in the array BNDLOC.

d. Output -

BNDLOC	Location of outer boundary of detailed particle traces

OUTP

a. Functions - This is utility subroutine to print out array information.

b. Input -

R	Array to be output
ISIZE	I dimension of R array
JSIZE	J dimension of R array
IBEG	Initial I value to be output
IEND	Final I value to be output
IINC	I value increment for output
JBEG	Initial J value to be output
JEND	Final J value to be output
JINC	J value increment for output
XSCALE	= 0.0 scale is calculated by subroutine to give best display. All elements or array are divided by scale before being output ≠ 0.0 scale is set by user
TITLE	Character string that identifies the array being output
NTP	Output unit number
NCOL	Number of columns of output device (usually either 80 or 132)

c. Processing - No other subroutines are called.

This subroutine takes any 2-dimensional array and formats the output.

d. Output - The array is output to the device NTP.

AURL1 (also FNDHDY, ADDHDY, FSUM)

a. Function - These subroutines determine the default Hardy precipitating electron flux.

b. Input -

FKP	Current Kp value
COLAT	Grid colatitude array (radians)
ALOCT	Time dependent magnetic local time grid (radians)
LATDIM	Number of latitudinal grid lines)
LTDIM	Number of local time (longitude) grid lines

c. Processing - following are the subprograms called and their major function:

FNDHDY	Subroutine to find and read the pertinent Fourier coefficients fot the Hardy electron model
ADDHDY	Subroutine to calculate the Hardy model electron flux value for a given point
FSUM	Function subprogram to determine sum of a Fourier series

These subroutines were supplied by Dr. David Hardy at the Air Force Geophysics Laboratory, Bedford, Mass.

d. Output -

AEFLUX	The integral electron energy flux (ergs/cm^2-sec)
AEMEAN	The average energy (eV)

WRT3D

a. Functions - This subroutine writes 3-d arravs to disk in the standard MSM disk file format.

b. Input -

LUN	Logical unit number on which to write file
IRECMX	Logical record number at which to write data
ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)
ARRAY	3-d array to be written
IDIM	I dimension of ARRAY
JDIM	J dimension of ARRAY
KDIM	K dimension of ARRAY
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine takes the desired array, opens the appropriate file, writes the array to the file starting with the specified logical record number, and closes the file.

d. Output - The array is written to unit number LUN.

RDHDR

a. Functions - This subroutine reads the header record of the standard MSM disk file

b. Input -

LUN	Logical unit number on which to write file
FILNAM	Character string containing file name
IRECMX	Record number to begin read function
IDIM	I dimension
JDIM	J dimension
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine opens the appropriate file, reads the header information of the file starting with the specified record number, and closes the file.

d. Output -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)

READ3D

a. Function - This subroutine reads a record from the standard MSM disk file.

b. Input -

LUN	Logical unit number from which to read file
FILNAM	Character string containing file name
LREC	Logical record number to read direct access data set
IDIM	1st dimension of ARRAY
JDIM	2nd dimension of ARRAY
KDIM	3rd dimension of ARRAY
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine opens the appropriate file, reads the array from the file starting with the specified record number, and closes the file.

d. Output -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)
ARRAY	3-d array to be read

SETALM

a. Functions - This subroutine reads the number of energy channels and their energies at geosynchronous orbit and calculates the energy invariant species corresponding to theses energies.

b. Input - Read from file 'ENCHAN'

IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
ENCHNL	Energy channel at geosynchronous orbit (eV)

c. Processing - No other subroutines are called.

The input data must be in eV with energy increasing within each chemical species. The parameter VM6 is the typical value for (flux tube volume)$^{-2/3}$ at geosynchronous orbit.

d. Output -

ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$
IEMAX	Maximum number of energy channels
IEDIM	Number of invariant energy species traced by the program
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
ALMDEL	Width of energy invariant channels

BNDSET

a. Functions - This subroutine computes the boundary plasma distribution.

b. input -

LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
ITMMAX	Actual number of time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
KPPLUS	Augmented KP dimension (=KPDIM+2)
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))
NAUGEL	Number of elements in the augmented input array
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
AUGPAR	Augmented data array for input values
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	(Flux tube volume)$^{-2/3}$ ((Re/nT)$^{-2/3}$)
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$
ALMDEL	Width of energy invariant channels
IEMAX	Maximum number of energy channels

c. Processing - No other subroutines are called.

The background for this subroutine is contained in Section 2.5.5 of Contract #F19628-87-0001 Final Report.

d. Output

ETABND	Boundary flux tube content distribution ($weber^{-1}$)
FLXBND	Flux values at boundary ($(cm^2\ sec)^{-1}$)

SETSCT

a. Functions - Subroutine to set species-dependent pitch angle scattering efficiency.

b. Input -

L	Time label
ISEC	Time (seconds)
IEDIM	Number of particle species to be traced
LTDIM	Number of local time (longitundinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
LATDIM	Number of latitundinal grid spaces
FKP	Kp value for current time
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$
ALOCT	Grid local time array (radians eastward from noon)
IEMAX	Maximum number of energy channels

c. Processing - No other subroutines are called.

This subroutine sets the species-dependent pitch angle scattering efficiency. The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

SCTFRC	Assumed pitch angle scattering efficiency

PFIX

a. Functions - This subroutine adjusts the modulus of a periodic J coordinate.

b. Input -

P1	Hour angle measured eastward from noon (radians)
P2	Hour angle array at position J (radians)
P3	Hour angle array at position J+1 (radians)

c. Processing - No other subroutines are called.

This subroutine adjusts the modulus of the periodic input parameters to match P1 so that they can be used to interpolate data.

d. Output -

P4	Adjusted hour angle array at position J (radians) (corresponds to P2)
P5	Adjusted hour angle array at position J+1 (radians) (corresponds to P3)

INITAL

a. Functions - This subroutine computes the initial plasma distribution.

b. Input -

LL	Time label
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/cm^2$-s-ster-eV))
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	(Flux tube volume)$^{-2/3}$ ((Re/nT)$^{-2/3}$)
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$
ALMDEL	Width of energy invariant channels
FKP	Kp at present time step
IEMAX	Maximum number of energy channels
AZERO	Statistical model weighting parameter
BZERO	Old results weighting parameter
CZERO	Traceback weighting paramater

c. Processing - following are the subroutines called and their major functions:

FLXVAL	Function subprogram to calculate Kp-dependent flux value a R=RR for particles of energy ENRG and Kp=FKP by interpolating FLXMAT array

This subroutine can compute the initial plasma distribution based on a combination of empirical (Kp-based) data and results from prefious runs. In practice, previous run results are used, if

available, and the empirical model is used otherwise. The rationale for this subroutine is discussed in Section 2.5.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

ETABEG Initial flux tube content distribution each time step ($weber^{-1}$)

TIMINC

a. Function - This subroutine increments the time for the next electric and magnetic field record.

b. Input -

ITIME	Current time (year, day, seconds)
IINC	Increment time (year,day,seconds)

c. Processing - No other subroutines are called.

This subroutine increments the time and checks that it is within bounds.

d. Output -

ITMNEW	Time for next record (year,day,seconds)

SETREF

a. Functions - This subroutine sets an upper limit reference flux.

b. Input -

LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	Flux tube volume ((Re/nT)$^{-2/3}$)
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$
ALMDEL	Width of energy invariant channels
IEMAX	Maximum number of energy channels
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
TVAL	Current time (seconds)
ITMMAX	Maximum number of time labels

c. Processing - following are the subroutines called and their major functions:

TNORML	Utility function subprogram to calculate run-normalized time
FLXVAL	Function subprogram to calculate Kp-dependent flux value a R=RR for particles of energy ENRG and Kp=FKP by interpolating FLXMAT array

The rationale for this subroutine is discussed in Section 2.5.6 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLXREF Upper limit to reference flux ($\log_{10}$(#/cm^2-s-ster-eV))

EFLOC

a. Functions - This subroutine finds the grid location (BI,BJ) of physical location (RVAL,XLT).

b. Input -

RVAL	Radial distance (Re)
XLT	Hour angle measured eastward from noon (radians)
LL	Time index
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
ALOCT	Grid local time array (radians eastward from noon)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)

c. Processing - following are the subroutines called and their major functions:

PFIX	Subroutine to adjust modulus of periodic J coordinate

This subroutine locate a point within a grid square and then linearly interpolates in I and J to give the non-integer locations.

d. Output -

BI	Non-integer I location of (RVAL,XLT)
BJ	Non-integer J location of (RVAL,XLT)

EFBNDY

a. Functions - This subroutine calculates the latitudes and widths of the electric field boundarys 2 and 3.

b. Input -

LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
LL	Time label
EQEDG	Equatorward edge of auroral oval (degrees)
STDOFF	Standoff distance (Re)
VDROP	Cross-polar cap potential drop
ALOCT	Grid local time array (radians eastward from noon)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
COLAT	Grid colatitude array (radians)

c. Processing - following are the subroutines called and their major functions.

EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT)
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

The rationale for this subroutine is discussed in Section 2.4.2 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Offset of ellipse in x(duskward) direction
DY	Offset of ellipse in y (duskward) direction

FECON

a. Functions - This subroutine obtains the time-normalized observational values.

b. Input -

ISTART	Start time (year,day, seconds)
IINC	Increment time (year,day,seconds)
IEND	Stop time (year,day,seconds)
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
ITMMAX	Number of time labels used
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

PARGEN	Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time
TIMINC	Subroutine to increment time for next electric- and magnetic field record
TILT	Subroutine to calculate the Earth's tilt angle
DSTDFL	Subroutine to calculate default (Kp driven) Dst value
EQTDFL	Subroutine to calculate default (Kp driven) equatorward edge of auroral zone value
STNDFL	Subroutine to calculate the standoff distance
PCPDFL	Subroutine to calculate default (Kp driven) cross-polar-cap potential value
PATDFL	Subroutine to return polar-cap convection pattern typ
CLPDFL	Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model

This subroutine calls Subroutine PARGEN to obtain time-normalized observational data that is then used to calculate other values needed by the electric field and magnetic field routines.

d. Output -

AUGPAR	Augmented data array for input values
MODE	Logical vector which gives whether each variable comes from actual data or a front end model.

PARGEN

a. Functions - Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time.

b. Input -

ISTART	Start time (year,day, seconds)
IINC	Increment time (year,day,seconds)
IEND	Stop time (year,day,seconds)
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated
NELTS	Number of elements in output data array

c. Processing - following are the subroutines called and their major functions:

INDATA	Subroutine to read in data for the MSM during development and testing.
SMOOTH	Subroutine to extract data from input array.
DTXIPT	Subroutine to return interpolated polar cap patterns
DTNTRP	Subroutine to return interpolated data values
TIMINC	Subroutine to increment time for next electric and magnetic field record

This subroutine retrieves observational data from the environmental data base and interpolates as necessary to have input data for the entire run. Mode is set to TRUE if there exists data for the run within the time constraints of array IGAP. If the gap in the data is too long, MODE is set false and the KP-based default models are used to compute the input data needed.

d. Output -

PARRAY	Data array of interpolated data values
MODE	Logical variable telling which input parameters we have values for

INDATA

a. Functions - Subroutine to read in data for the MSM during development and testing.

b. Input -

PARAM	File name to read input data from (character)
STARTT	Start time (year,day,seconds)
ENDT	End time (year,day,seconds)
NDIM	Maximum number of data points to read

c. Processing - No other routines are called.

This routine opens a data file and reads the observation data. This routine will be replaced in the operational MSM with an interface routine to the AWS environmental data base.

d. Output -

DARRY	Array containing observational data
NUMNUM	Number of data points read

GETMAT

a. Functions - Subroutine to find the magnetic field matrices needed for interpolation in order to represent current geophysical conditions.

b. Input -

IWANT	Different magnetic field parameters = 1 - XMIN (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 2 -YMIN (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 3 - ZMIN (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 4 - BMIN (Equatorial magnetic field strength (nT)) = 5 - VM (Flux tube volume (weber^{-1}))
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
BFPAR	Array containing geophysical parameters to use to find magnetic field matrices 1 = Standoff distance (Re) 2 = Tilt angle (degrees) 3 = Equatorward edge of auroral oval (degrees) 4 = Dst (nT) 5 = Collapse parameter

c. Processing - following are the subroutines called and their major functions:

FNDBRK	Subroutine to find the proper magnetic field indces that bracket the magnetic field input parameters
LOADBM	Subroutine to load individual magnetic field matrices from the magnetic field super matrix into the work magnetic field matrices
ZEROBM	Subroutine to zero out an individual magnetic field matrix within the working magnetic field matrices
OUTP	Utility subroutine to print out array information

d. Output -

WORK	Internal working array
BFLIM	Indices of magnetic field models read
BFEXST	Array of logical variables of which magnetic field model matrices exist

FNDBRK

a. Functions - Subroutine to find the magnetic field array indices that bracket the magnetic field input parameters.

b. Input -

PARVAL	Magnetic field geophysical parameter
PVALS	Array of magnetic field parameters
IPDIM	Dimension of PVALS

c. Processing - No other subroutines are called.

This routine finds the indices of the values in array PVAL that bracket the input geophysical parameter.

d. Output -

MIN	Lower array bound index
MAX	Upper array bound index

MEXIST

a. Functions - Logical functin that checks to see if a particular B-matrix exists in the B supermatrix

b. Intput -

MSTND	Number of standoff parameters, currently set to 5
MTILT	Number of tilt angle parameters, currently set to 5
MINED	Number of auroral equatorward edge parameters, currently set to 16
MDST	Number of Dst parameters, currently set to 8
MSTCH	Number of tail collapse parameters, currently set to 2

c. Processing - No other subroutines are called.

This function uses the FORTRAN "INQUIRE" command to check for the existence of needed B-field arrays/

d. Output -

MEXIST	true if B field arrays exist, false otherwise

ZEROBM

a. Functions - Subroutine to zero out an individual magnetic field matrix within the working magnetic field matrices if the matrix does not exist in the supermatrix.

b. Input -

LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
I,J,K,L,M	Indices of working magnetic field matrices
WORK	Working magnetic field matrix

c. Processing - No other subroutines are called.

So that the program does not crash if a magnetic field matrix does not exist, it is zeroed out. The printout within the program prints that no magnetic field matrix existed for those geophysical conditions.

d. Output -

WORK	Zeroed out working magnetic field matrix

LOADBM

a. Processing - Subroutine to load individual matnetic field matrices from the magnetic field super-matrix into the work magnetic field matrices.

c. Input -

IWANT	5 different magnetic field parameters = 1 - XMIN (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 2 -YMIN (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 3 - ZMIN (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 4 - BMIN (Equatorial magnetic field strength (nT)) = 5 - VM ((Flux tube volume)$^{-2/3}$ ((Re/nT)$^{-2/3}$))
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
I,J,K,L,M	Indices of working magnetic field matrices
BFNDX	Array containing the indices of the magnetic field matrices to load for geophysical conditions of standoff distance, tilt angle, equatorward edge of the auroral oval, Dst, and tail collapse index

c. Processing - No other subroutines are called.

This subroutine loads individual magnetic field matrices from the offline magnetic field super-matrix into the work magnetic field matrices. A check is made to verify that the correct matrix has been retrieved. It this check fails the program will stop.

d. Output -

Work	Magnetic field matrix work array

TCONV2

a. Functions - Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.

b. Input -

YR	Current year
DD	Current day
STRTYR	Start year
STRTDD	Start day

c. Processing - No other subroutines are called.

d. Output -

TCONV2	Seconds from midnight of start day

TCONV3

a. Functions - Function to take a start year, start day, current year, current day, and current seconds of day to produce a value that is the number of seconds from midnight of the start year and day.

b. Input -

ITIME	Current time (year,day,seconds)
ISTART	Start time (year,day,seconds)

c. Processing - No other subroutines are called.

d. Output -

TCONV3	Seconds from the start year and day

SMOOTH

a. Functions - Subroutine to extract and interpolate data from observational data into a usable form for the MSM.

b. Input -

DARRY	Array containing input observational data
NDIM	Maximum number of data points in DARRY
NUMNUM	Number of data points in DARRY
STARTY	Start year
STARTD	Start day

c. Processing - following are the subroutines called and their major functions:

TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.

This subroutine extracts the data relevant to the total run time.

d. Output -

NA	Total number of data points found
XA	Time (seconds)
YA	Data point
IERR	Error code = 0 no errors detected = -1 error in data

DTNTRP

a. Functions - Subroutine to return interpolated data.

b. Input -

X	Time (seconds)
NA	Number of data points
XA	Time array of observational data (seconds)
YA	Array of observed data

c. Processing - No other subroutines are called.

This subroutine takes the given time, finds the bracketing time in the XA array and linearly interpolates the observational data. It also does the differential DY/DX.

d. Output -

Y	Interpolated observational data
DYDX	Differential DY/DX
DELTA	The difference between the given time and the time of the observational data
IERR	Error code = 0 no problems in interpolation = -1 error detected

STNDOF

a. Functions - Subroutine to calculate the standoff distance from solar wind velocity and density data.

b. Input -

VELOC	Solar wind velocity data (km/sec)
DENS	Solar wind density data (cm^{-3})

c. Processing - No other subroutines are called.

This is based on the analytic calculation of Alpbach (1979).

d. Output -

STAND	Standoff distance (Re)

FLXNIT

a. Functions - Subroutine to calculate Kp based particle number flux as function of R, energy and Kp.

b. Input -

IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated

c. Processing - following are the subroutines called and their major functions:

FLXCAL	Subroutine to compute Kp-dependent flux at L=3, 4, 6.6, and 13 for electrons of given energy for a given Kp condition
THRCAL	Subroutine to calculate threshold flux at R=3, 4, 6.6, and 13 Re for given Kp and energy

This subroutine sets up the threshold and default fluxes for the program run.

d. Output -

FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. ($\log_{10}$(#/cm^2-s-ster-eV))

FLXCAL

a. Functions - Subroutine to compute Kp-dependent flux at L=3, 4, 6.6 and 13 for different species of given energy for a given Kp condition.

b. Input -

KPDIM	Number of KP values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
FNRGLG	One energy value for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLKP	One KP value for which FLXMAT arrays are calculated
ISPP	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.5 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLX3	Flux at L=3 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX4	Flux at L=4 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX6	Flux at L=6 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX13	Flux at L=13 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp

FLXTRP

a. Functions - Function subprogram to return Log_{10} of empirical flux value interpolated between values at L=3, 4, 6.6, and 13.

b. Input -

RLOG	Log_{10} of radial distance to be interpolated
FLX3	Flux at L=3 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX4	Flux at L=4 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX6	Flux at L=6 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp
FLX13	Flux at L=13 (#/cm^{-2}-sec-ster-eV) for given energy for given Kp

c. Processing - No other subroutines are called.

d. Output -

FLXTRP	Flux value interpolated in radial distance

FLXVAL

a. Functions - Function subprogram to calculate Kp-dependent flux value for particles at a given radial distance, energy, and Kp conditions by interpolating the FLXMAT array.

b. Input -

ISP	Vector of chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
FKP	Kp for current time
RR	Radial distance of particle (Re)
PP	Hour angle (measured eastward from noon in radians) in equatorial plane
ENRG	Energy (eV)
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FLXKP	KP values for which FLXMAT arrays are calculated

c. Processing - following are the subroutines called and their major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

This subroutine performs a general 3-d interpolation in radial distance, Kp, and energy.

d. Output -

FLXVAL	Interpolated flux

EFIELD

a. Functions - Subroutine to return values of the potential on the northern hemisphere grid, southern hemisphere grid, and the average of the two hemispheres.

b. Input -

KPATT	Polar cap pattern type = 0 Bz > 0 = 1 Bz < 0 with stronger flow on dawn side of polar cap = 2 Bz < 0 with symmetric flow in polar cap = 3 Bz < 0 with stronger flow on dusk side of polar cap
PCP	Cross polar cap potential drop (kV)
DEQDT	Time rate of change of the equatorward edge of the auroral zone (degrees/hour)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere offset of ellipse measured in x(sunward) direction
DYS	Southern hemisphere offset of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere offset of ellipse measured in x (sunward) direction
DYN	Northern hemisphere offset of ellipse measured in y (duskward) direction
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
MODE	Logical variable telling which input parameters we have values for
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
JWRAP	Number of points of overlap in J direction
ITMCUR	Current time label
ITMDIM	Maximum number of major time steps per run of the program
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes , but scaled to externally specified cross polar cap potential drop and ellipse parameters

= -1 means Heppner-Maynard formula is used for all latitudes, unscaled.
= -2 should not be used. It is the form used by the Rice Convection Model

ITOP = 1 means the ellipse parameters A, B, DX, and DY for the polar cap boundary are computed internally in the subroutine. This is the normal mode for the MSM.

c. Processing - following are the subroutines called and their major functions:

EMODEL Returns the electric potential arrays on the northern hemisphere grid, southern hemisphere grid, and the average of the two hemispheres

The major function of this subroutine is to translate the KPATT pattern types given by the University of Texas, Dallas to the interally used IPATT pattern types of the Heppner-Maynard empirical model. It also keeps track of which hemisphere is being modelled and averages the output from the two hemispheres.

d. Output -

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
VNORTH	Northern hemisphere electric potential distribution (Volts)
VSOUTH	Southern hemisphere electric potential distribution (Volts)

PWRCAL

a. Functions - Subroutine to estimate precipitating leectron energy flux poleward of the main MSM modeling region by comparing flux within the modeling region with AFGL statistical values.

b. Input -

LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
ISPDIM	Number of plasma mass species (electrons, H^+, O^+, etc.)
ITMCUR	Current time label
ITMDIM	Maximum number of major time steps per run of the program
FKP	Kp for current time
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
ALPHA	Latitundinal grid spacing vector
BETA	Longitudinal grid spacing vector
BNDLOC	Location of outer boundary of detailed particle traces
A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Offset of ellipse measured in x (sunward) direction
DY	Radius of ellipse measured in y (duskward) direction

c. Processing - following are the subroutines called and their major functions:

THET	Function that gives equation for and electric field boundary ellipse in flat polar coordinates

The rationale for this subroutine is discussed in Section 2.6.5.2 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H^+ and (3) O^+) (ergs/cm^2-sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H^+ and (3) O^+(eV)

WKRATE

a. Functions - Function subroutine to evaluate weak precipitation loss rate.

b. Input -

RMID	R value at point where function is to be evaluated (Re)
RPPMID	Estimate of plasmapause radius at midnight local time(Re)
PMID	Local time hour angle in equatorial plane (radians from local noon)
ENRG	Particle energy (eV)

c. Processing - following are the subroutines called and their major functions:

RLYONS	Function to evaluate Lyons-based weak loss rate

The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

WKRATE	Estimate of weak precipitation loss rate

RLYONS

a. Functions - Function to evaluate Lyons-based weak loss rate.

b. Input -

RMID	R value at point where function is to be evaluated (Re)
ENRG	Particle energy (eV)

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

RLYONS	Estimate of weak precipitation loss rate from L. Lyons formula

G3TRPA

a. Functions - Utility function subprogram to perform a general 3-d linear interpolation of an angular array A(I,J,K) at point (BI,BJ,BK)

b. Input -

A	Array to be interpolated
IMAX	I dimension of array A
JMAX	J dimension of array A
KMAX	K dimension of array A
BI	Floating point value to interpolate in I dimension
BJ	Floating point value to interpolate in J dimension
BK	Floating point value to interpolate in K dimension

c. Processing - No other subroutines are called.

This subroutine uses the standard linear interpolation technique to interpolate an angular array in 3 dimensions.

d. Output -

G3TRPA	Interpolated point from array A

THRCAL

a. Functions - Subroutine to calculate threshold flux at R=3, 4, 6.6, and 13 Re for given Kp value and energy.

b. Input -

FNRGLG	One energy value for which FLXMAT arrays are calculated ($\log_{10}$(eV))
FKP	One KP value for which FLXMAT arrays are calculated
NRGNDX	Energy calculation index
KPNDX	Kp calculation index
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}$(#/cm^2-s-ster-eV))

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.5.8 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

THR	Value for threshold flux (particles/cm^2-s-ster-eV)

FLXDFL

a. Functions - Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array.

b. Input -

ITM	Current time label
FKP	Current Kp
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	Flux tube volume $((Re/nT)^{-2/3})$
ALAM	Energy invariant vector $(eV(Re/nT)^{2/3})$
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values $(\log_{10}(\#/cm^2\text{-s-ster-eV}))$
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated $(\log_{10}(eV))$
FLXKP	KP values for which FLXMAT arrays are calculated
ALMDEL	Width of energy invariant channels
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
ITMDIM	Maximum number of major time steps per run of the program
BNDLOC	Location of outer boundary of detailed particle traces
IEMAX	Maximum number of energy channels
IEDIM	Number of invariant energy species traced by the program
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
IEBEG	Beginning energy index
IEEND	Ending energy index

c. Processing - following are the subroutines called and their major functions:

FLXVAL	Function subprogram to calculate Kp-dependent flux value at given radial distance, energy and Kp by interpolating FLXMAT array.

This routine is used as the MSM default model when full particle traces are done. The subroutine interpolates the FLXMAT arrays for the given energies for all grid points.

d. Output -

ETA	Flux tube content array (weber^{-1})
EFLUX	Precipitation energy flux array (ergs/cm^2-sec)

RMVBSH

a. Functions - Subroutine to extrapolate magnetic field matrices into extremely stretched or open regions.

b. Input -

WORK	Working magnetic field matrix
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces

c. Processing - No other subroutines are called.

To prepare the magnetic field matrices for interpolation, previously assigned 'very large numbers' to signify that the magnetic field extended beyond 50 Re are removed. The subroutine looks for the closest value in local time (J) and linearly interpolates for the latitudinal (I) grid points.

d. Output -

WORK	Revised working magnetic field matrix

DSTDFL

a. Functions - Subroutine to calculate default (Kp-driven) Dst value

b. Input -

MODE	Logical variable telling which input parameters we have values for
AUGPAR	Augmented data array for input values
NAUGEL	Number of elements in the augmented input array
ITMDIM	Maximum number of major time steps per run of the program
ITM	Current time label
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

DTNTRP	Subroutine to return interpolated data values

The subroutine uses the Kp value in the AUGPAR array for the given time and calculates Dst. This value is then put into the AUGPAR array. Subroutine DTNTRP is called to give the DY/DX value also needed in AUGPAR.

d. Output -

AUGPAR	Augmented data array for input values

EQTDFL

a. Function - Subroutine to calculate default (Kp-driven) equatorward edge of the auroral oval value.

b. Input -

MODE	Logical variable telling which input parameters we have values for
AUGPAR	Augmented data array for input values
NAUGEL	Number of elements in the augmented input array
ITMDIM	Maximum number of major time steps per run of the program
ITM	Current time label
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

DTNTRP Subroutine to return interpolated data values

The subroutine uses the Kp value in the AUGPAR array for the given time and calculates the location of the equatorward edge of the auroral oval. The algorithm is given in a JGR paper by Gussenhoven et al. (1983). This value is then put into the AUGPAR array. Subroutine DTNTRP is called to give the DY/DX value also needed in AUGPAR.

d. Output -

AUGPAR Augmented data array for input values

PDPDFL

a. Functions - Subroutine to calculate default (Kp-driven) cross-polar-cap potential value.

b. Input -

MODE	Logical variable telling which input parameters we have values for
XKP	Current KP value

c. Processing - No other subroutines are called.

This subroutine uses the algorithms of Reiff et al to calculate the cross-polar-cap potential from given Kp.

d. Output -

PCP	Cross-polar cap potential (kV)

PATDFL

a. Functions - Subroutine to return polar cap convection pattern type of the basis of the measurements of the By and Bz components of the interplanetary magnetic field.

b. Input -

MODE	Logical variable telling which input parameters we have values for
MODEBY	Logical variable telling if we have By data
MODEBZ	Logical variable telling if we have Bz data

c. Processing - No other subroutines are called.

This subroutine determines the pattern type using interplanetary magnetic field data:
Polar cap pattern type
= 0 Bz > 0
= 1 Bz < 0 with stronger flow on dawn side of polar cap
= 2 Bz < 0 with symmetric flow in polar cap
= 3 Bz < 0 with stronger flow on dusk side of polar cap
If no data is available, pattern #2 is the default.

d. Output -

IPATT	Polar cap pattern type

CLPDFL

a. Functions - Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model.

b. Input -

MODE	Logical variable telling which input parameters we have values for

c. Processing - No other subroutines are called.

If mode is false, no value has been returned from the environmental data base and the default value is set to 1 (no collapse).

d. Output -

CLAPSE	Value of magnetotail collapse parameter

STNDFL

a. Functions - Subroutine to calculate the standoff distance.

b. Input -

MODE	Logical variable telling which input parameters we have values for
MODVEL	Logical variable telling if we have solar wind velocity data
MODDEN	Logical variable telling if we have solar wind density data
SWVEL	Value of solar wind velocity data
SWDEN	Value of solar wind density data
XKP	Current Kp value

c. Processing - following are the subroutines called and their major functions:

STNDOF	Subroutine to calculate the standoff distance according to a formula published by Alpbach (1979)

If solar wind velocity and density data are available Subroutine STNDOF does the actual calculation. Otherwise the default is a Kp-based formula.

d. Output -

STAND	Standoff distance (Re)

TILT

a. Functions - Subroutine to calculated the earth's tilt angle.

b. Input -

DAY	Day of year
YEAR	Year

c. Processing - No other subroutines are called.

This subroutine currently sets the tilt angle at 0.0.

d. Output -

XTILT	Tilt angle

AURL2S

a. Function - This subroutine determines the default Hardy precipitating ion flux.

b. Input -

FKP	Current Kp value
COLAT	Grid colatitude array (radians)
ALOCT	Time dependent magnetic local time grid
LATDIM	Number of latitudinal grid lines)
LTDIM	Number of local time (longitude) grid lines

c. Processing - following are the subroutines called and their major function:

REGEN	Subroutine used in evaluation of precipitatin ion fluxes
EFUN	Subroutine used in evaluation of precipitatin ion fluxes

These set of subroutines were developed by Dr. David Hardy at the Air Force Geophysics Laboratory, Bedford, Mass.

d. Output -

AEFLUX	The integral ion energy flux (ergs/cm^2-sec)
AEMEAN	The average energy (eV)

OPTERR

a. Functions - Subroutine to calculate the error and standard deviation between the MSM flux and observational geosynchronous satellite data.

b. Input -

FLUX	Particle flux array $(cm^2\ s)^{-1}$
VM	(Flux tube volume)$^{-2/3}$ ((Re/nT)$^{-2/3}$)
LATDIM	Number of latitundinal grid spaces
LTDIM	Number of local time (longitundinal) grid spaces
IEDIM	Maximum number of energy channels
IEMAX	Number of invariant energy species traced by the program
ALAM	Energy invariant vector (eV(Re/nT)$^{2/3}$)
ALMDEL	Width of energy invariant channels
THRSH	Threshold energy beyond which particles are not traced and empirical data are used
ITMDIM	Maximum number of major time steps per run of the program
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
ISTART	Start time (year,day,seconds)
IEND	End time (year,day,seconds)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
TIMTAG	Vector giving times at which E and B parameters are calculated
ITMMAX	Number of time labels in run

c. Processing - following are the subroutines called and their major functions:

TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day
READ3D	Subroutine to read a record from the standard MSM disk file format
TNORML	Utility function subprogram to calculate run-normalized time
EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT)
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

For the satellite times and energies available, the subroutine calculates and interpolates the model flux values at geosynchronous orbit. The error is defined as the average difference between the log of the model flux and the log of the satellite flux. The standard deviation is the usual definition. The subroutine averages over all available satellites. The format of the file 'EPSAT' will need to be changed when the MSM begins to access the AWS environmental data base. The standard deviation is not saved, just printed out.

d. Output -

ERSHFT	Vector of error values

WRTVEC

a. Function - This subroutine writes to disk the energy invariant, the threshold energies, and the error shift for each particle energy species each time step.

b. Input -

NOUNIT	Logical unit number from which to read file
IRDOUT	Output direct access record number
IEMAX	Number of particle energy species for run
IEDIM	Dimension of particle energy species array
ALAM	Vector of energy invariant values used in run
THRSH	Vector of full-traceback/default value used in run
ERSHFT	Vector of error values for this time interval
IFLAV	Vector of chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions

c. Processing - No other subroutines are called.

This routine opens the appropriate file, writes the vectors to the file, and closes the file.

d. Output - The vectors are output to unit number NOUNIT.

CEXRAT

a. Function - Function subprogram to return charge exchange loss rate (sec^{-1}) for ions of species ISP, energy ENRG (eV), at L=RLOC (Re) for sunspot number SSN. This routine is based on a table generated by James Bishop of the University of Michigan.

b. Input -

ISP	Species Identifier
	ISP=2 for H^+ Ions
	ISP=3 for O^+ Ions
ENRG	Energy in eV
RLOC	Radial location (Re)
SSN	Sunspot number
DKTIME	Table of ion decay times
IRDK	Radial dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sunspot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

c. Processing - following are the subroutines called and their major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

This routine interpolates the decay time array in radial distance, energy and sunspot number.

d. Output -

CEXRAT	Ion charge exchange loss rate (sec^{-1})

READDK

a. Function - Subroutine to read ion charge exchange decay table furnished by James Bishop of the University of Michigan.

b. Input -

IRDK	Radial dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sunspot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

c. Processing -

Reads the input table of ion decay times.

d. Output -

DKTIME	Table of ion decay times

FLX2ET

a. Function - Convert from particle flux (particles/cm^2/s) to invariant density eta (particles/Weber)

b. Input -

LATDIM	Number of latitudinal grid points
LTDIM	Number of longitudinal (local time) grid points
IEDIM	Maximum number of model energy channels
IEMAX	Actual number of model energy channels
ITMDIM	Maximum time depth
ITM	Time index
IFLAV	Vector giving charge and mass species of particles 1 = electrons, 2 = H^+ ions, 3 = O^+ ions
ALAM	Energy invariant vector
ALMDEL	Width of invariant energy channels
BNDLOC	Boundary location array
VM	(Flux tube volume)$^{-2/3}$
FLUX	Particle number flux (#/cm^2/s)

c. Processing - No other subroutines are called.

This subroutine uses a simple formula to translate from number flux FLUX to invariant density ETA.

d. Output -

ETA	Invariant number density (particles / Weber)

Appendix B

ET2FLX

a. Functicn - Convert from invariant density eta (particles/Weber) to particle number flux (particles/cm^2/s)

b. Input -

LATDIM	Number of latitudinal grid points
LTDIM	Number of longitudinal (local time) grid points
IEDIM	Maximum number of model energy channels
IEMAX	Actual number of model energy channels
ITMDIM	Maximum time depth
ITM	Time index
IFLAV	Vector giving charge and mass species of particles 1 = electrons, 2 = H^+ ions, 3 = O^+ ions
ALAM	Energy invariant vector
ALMDEL	Width of invariant energy channels
BNDLOC	Boundary location array
VM	(Flux tube volume)$^{-2/3}$
ETA	Invariant number density (particles / Weber)

c. Processing - No other subroutines are called.

This subroutine uses a simple formula to translate from invariant density ETA to number flux FLUX.

d. Output -

FLUX	Particle number flux (#/cm^2/s)

```
$ define bfield [.bfields] /process
$ define for005 msmin.dat /process
$ define for006 print3.dat /process
$ define for009 errorl.dat /process
$ define for010 efield.coeff /process
$ define for011 bfield.coeff /process
$ define for012 parray.dat /process
$ define for013 pieng.dat /process
$ define for015 print.dat /process
$ define for018 tt /process
$ define for020 dktable.dat /process
$ define for021 etabnd.dat /process
$ define for023 shift.dat /process
$ define for024 piflx.dat /process
$ define for025 eavg.dat /process
$ define for026 flxsum.dat /process
$ define for027 flux.dat /process
$ define for035 v.dat /process
$ define for036 vm.dat /process
$ define for037 bmin.dat /process
$ define for038 xmin.dat /process
$ define for039 ymin.dat /process
$ define for040 zmin.dat /process
$ define for041 aloct.dat /process
$ define for042 colat.dat /process
$ define for043 bndloc.dat /process
$ define for044 updat.dat /process
$ define for045 vnorth.dat /process
$ define for046 vsouth.dat /process
$ define efcoef efcoef.dat /process
$ define coord  coord.dat /process
$ define hardy  hardy.dat /process
$ define epsat  epsat.dat /process
$ define epions epions.dat /process
$ define ioneng ioneng.dat /process
$ define ionnum ionnum.dat /process
$ define enchan enchan.dat /process
$ define fkp    fkp.dat /process
$ define dst    dst.dat /process
$ define eqedge eqedge.dat /process
$ define clapse clapse.dat /process
$ define swden  swden.dat /process
$ define swvel  swvel.dat /process
$ define pcp    pcp.dat /process
$ define xipatt xipatt.dat /process
$ run msml
```

```
      PROGRAM MSMCON
C
C  VERSION 1.0                 DATE: NOVEMBER 10, 1988
C          1.1                          MARCH 26, 1990
C          1.2                            MAY 11, 1990
C          1.3                           JULY 01, 1990
C
C  PROGRAMMER: BOB SPIRO
C
C  PURPOSE:  MSMCON IS THE MAIN PROGRAM FOR THE MAGNETOSPHERIC
C          SPECIFICATION MODEL.  MSMCON CALLS THE SUBROUTINES PPTCON,
C          HIEPAR, AND OUTPUT.
C
         PARAMETER (LATDIM=62,LTDIM=51,IEDIM=30,ITMDIM=50,
     2               NAUGEL=28,IRDIM=4,NRGDIM=29,KPDIM=7,ISPDIM=3,
     3               IRDK=18,INRGDK=13,ISOLDK=2,IONDK=2,KPPLUS=9)
C
C  LATDIM GIVES THE NUMBER OF LATITUINAL GRID LINES (ALIASED AS IDIM
C         IN SOME ROUTINES)
C  LTDIM GIVES THE NUMBER OF LOCAL TIME (LONGITUDE) GRID LINES (ALIASED
C         AS JDIM IN SOME ROUTINES)
C  IEDIM GIVES THE NUMBER OF INVARIANT ENERGY SPECIES TRACED BY THE
C         PROGRAM
C  ITMDIM GIVES THE MAXIMUM NUMBER OF MAJOR TIME STEPS PER RUN OF THE
C         PROGRAM
C  NAUGEL GIVES THE NUMBER OF ELEMENTS IN THE AUGMENTED INPUT ARRAY
C
C  IRDIM GIVES THE NUMBER OF R VALUES FOR WHICH THE EMPIRICAL FLUX
C        ARRAY, FLXMAT, IS CALCULATED
C  NRGDIM GIVES THE NUMBER OF PRESET ENERGY VALUES FOR WHICH FLXMAT IS
C        CALCULATED
C  KPDIM GIVES THE NUMBER OF KP VALUES FOR WHICH FLXMAT IS CALCULATED
C
C  KPPLUS GIVES THE KP DIMENSION OF THE AUGMENTED FLXMAT ARRAY
C   (KPDIM + 2)
C
C  ISPDIM GIVES THE NUMBER OF PLASMA MASS SPECIES (ELECTRONS,H+,O+,ETC)
C
C
      CHARACTER*80 CHID
C
C  COMMON LUNIT CONTAINS LOGICAL UNIT NUMBERS FOR INPUT AND OUTPUT
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C  COMMON IRDREC CONTAINS RECORD NUMBERS FOR PARTICE TRACE (IRECPT)
C      AND E AND B FIELD FILES (IRECEB)
         COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG
C
C  COMMON SUN CONTAINS THE SUN SPOT NUMBER FOR THE ION CHARGE
C      EXCHANGE ALGORITHM
         COMMON /SUN/ SSN
C
C  WITH FEW EXCEPTIONS, MOST OF THE ARRAYS USED IN THE PROGRAM ARE
C    DIMENSIONED HERE IN MSMCON.
C
      DIMENSION ISTART(3),IINC(3),IEND(3)
      DIMENSION ETA(LATDIM,LTDIM,IEDIM),EFLUX(LATDIM,LTDIM,IEDIM),
     2    FLXSUM(LATDIM,LTDIM,3),EAVG(LATDIM,LTDIM,3),
     3    FLUX(LATDIM,LTDIM,IEDIM)
C
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),V(LATDIM,LTDIM,ITMDIM),
```

```
     1     ETABEG(LATDIM,LTDIM,IEDIM),ETABND(LTDIM,IEDIM,ITMDIM),
     2     FLXBND(LTDIM,IEDIM,ITMDIM),
     3     BNDLOC(LTDIM,ITMDIM),ALAM(IEDIM),COLAT(LATDIM,LTDIM),
     4     ALOCT(LATDIM,LTDIM),BMIN(LATDIM,LTDIM),XMIN(LATDIM,LTDIM),
     5     YMIN(LATDIM,LTDIM),ZMIN(LATDIM,LTDIM),TIMTAG(ITMDIM),
     6     ALMDEL(IEDIM)
C
      DIMENSION VNORTH(LATDIM,LTDIM),VSOUTH(LATDIM,LTDIM)
C
      DIMENSION IFLAV(IEDIM),BNDMAX(LTDIM),
     1  ALPHA(LATDIM),BETA(LATDIM),BIR(LATDIM,LTDIM),
     2  ASOUTH(3,ITMDIM),BSOUTH(3,ITMDIM),DXS(3,ITMDIM),DYS(3,ITMDIM),
     3  ANORTH(3,ITMDIM),BNORTH(3,ITMDIM),DXN(3,ITMDIM),DYN(3,ITMDIM),
     4  VMLOSS(LATDIM,LTDIM,ITMDIM),
     5  TETA(LATDIM),PHI(LTDIM),SINI(LATDIM,LTDIM)
C
      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)
C
      DIMENSION SCTFRC(IEDIM,LTDIM,ITMDIM),RPP(ITMDIM)
C
      DIMENSION DKTIME(IRDK,INRGDK,ISOLDK,IONDK)
C
C
      DIMENSION ID(20),RID(20),THRSH(IEDIM),ERSHFT(IEDIM)
C
C
      DIMENSION AUGPAR(NAUGEL,ITMDIM)
      LOGICAL*1 MODE(NAUGEL,ITMDIM)
C
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)
      DIMENSION THRMAT(IRDIM,NRGDIM,KPDIM)
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)
C
C  CALL PRECIPITATION AND PARTICLE TRACER CONTROL
C
C
C  INPUT MAJOR TIME STEP INCREMENT (YEARS, DAYS, SECONDS)
        IINC(1) = 0
        IINC(2) = 0
        IINC(3) = 900
C
C READ IN START YEAR, START DAY, START TIME IN SECONDS
      READ(5,*) IYEAR,IDAY,ISEC
      ISTART(1)=IYEAR-(IYEAR/100)*100
      ISTART(2)=IDAY
      ISTART(3)=ISEC
C
C  READ IN STOP YEAR, STOP DAY, STOP TIME IN SECONDS
      READ(5,*) IYEAR,IDAY,ISEC
      IEND(1)=IYEAR-(IYEAR/100)*100
      IEND(2)=IDAY
      IEND(3)=ISEC
C
C  READ IN START RECORD NUMBER
      READ(5,*) IRECEB
      IRECPT=IRECEB
C
C  READ IN STRING TO IDENTIFY RUN
      READ(5,*) CHID
C
C  READ IN SUN SPOT NUMBER SSN
      READ(5,*) SSN
C
      WRITE(6,*) 'ISTART=',ISTART
```

```
      WRITE(6,*) 'IEND=',IEND
      WRITE(6,*) 'IRECPT=',IRECPT,'IRECEB=',IRECEB
      WRITE(6,*) 'CHID=',CHID
      WRITE(6,*) 'SSN=',SSN
C
         IRDBEG=IRECPT
C
C
C     INITIALIZE TIMER
C
C  CALLS TO DTIME ARE PLACED IN THE CODE TO PROVIDE TIMING INFORMATION
C    FOR DIFFERENT SECTIONS OF THE PROGRAM.  THE DUMMY SUBROUTINE STUB
C    DTIME NEEDS TO BE REPLACED WITH MACHINE-SPECIFIC TIMING CALLS IN
C    ORDER TO OBTAIN REAL TIMING INFORMATION.
C
c        CALL DTIME(ITODST)
C
C  CALL TO PPTCON READS AND ASSIMILATES AVAILABLE INPUT DATA, PRODUCES
C    E AND B FIELD MATRICES FOR PERIOD OF INTEREST, AND COMPUTES
C    PARTICLE FLUXES IN MAGNETOSPHERE AND PRECIPITATING INTO
C    IONOSPHERE.
C
       CALL PPTCON(ISTART,IINC,IEND,LATDIM,LTDIM,IEDIM,ITMDIM,
     1   IRDK,INRGDK,ISOLDK,IONDK,
     1   IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
     2   THRMAT,CHID,ID,RID,SCTFRC,IFLAV,BNDMAX,ALPHA,BETA,TETA,PHI,BIR,
     3   SINI,TIMTAG,COLAT,ALOCT,V,VNORTH,VSOUTH,ASOUTH,BSOUTH,DXS,DYS,
     4   ANORTH,BNORTH,DXN,DYN,VM,VMLOSS,BMIN,XMIN,YMIN,ZMIN,
     5   ALAM,ETABEG,ETABND,FLXBND,BNDLOC,ETA,FLUX,EFLUX,FLXSUM,
     6   EAVG,R,P,
     7   ALMDEL,NAUGEL,AUGPAR,MODE,RPP,THRSH,ERSHFT,DKTIME)
C
C     FIND TIME TO EXECUTE PPTCON
C
c        CALL DTIME(IPPTIM)
c        WRITE (LUERR,*) 'THE TIME SPENT IN PPTCON IS ',
c    1               ABS(IPPTIM-ITODST)/100.,' SECONDS.'
C
C  CALL HIEPAR
C    CALL TO HIEPAR GIVES THE STATISTICS-BASED HIGH ENERGY (> 300 KEV)
C    OUTPUT OF THE PROGRAM.  THE JULY 1, 1990, VERSION OF THE MSM FOR
C    DELIVERY TO THE AIR FORCE CONTAINS ONLY A DUMMY STUB FOR HIEPAR.
C
         CALL HIEPAR
C
C     FIND TIME TO EXECUTE HIEPAR
C
c        CALL DTIME(IHIETM)
c        WRITE (LUERR,*) 'THE TIME SPENT IN HIEPAR IS ',
c    1               ABS(IHIETM-IPPTIM)/100.,' SECONDS.'
C
C  CALL OUTPUT
C    IN THE SFC OPERATIONAL VERSION OF THE MSM, THE CALL TO OUTPUT
C    WILL OUTPUT MSM RESULTS TO THE ENVIRONMENTAL DATA BASE.
C    THE JULY 1, 1990, DELIVERY VERSION OF THE MSM CONTAINS A STUB FOR
C    SUBROUTINE OUTPUT.
C
         CALL OUTPUT
C
C     FIND TIME TO EXECUTE OUTPUT
C
c        CALL DTIME(IOUTIM)
c        WRITE (LUERR,*) 'THE TIME SPENT IN OUTPUT IS ',
c    1               ABS(IOUTIM-IHIETM)/100.,' SECONDS.'
```

```
C                                                                       MSM01930
C     FIND THE TOTAL TIME OF THE RUN                                    MSM01940
C                                                                       MSM01950
c        CALL DTIME(ITODED)                                             MSM01960
c        WRITE (LUERR,*) 'THE TOTAL TIME SPENT IN MSMCON IS ',          MSM01970
c    1                   ABS(ITODED-ITODST)/100.,' SECONDS.'            MSM01980
C                                                                       MSM01990
      STOP                                                              MSM02000
      END                                                               MSM02010
C                                                                       MSM02020
C **************************** P P T C O N *************************    MSM02030
C                                                                       MSM02040
      SUBROUTINE PPTCON(ISTART,IINC,IEND,LATDIM,LTDIM,IEDIM,ITMDIM,     MSM02050
     1     IRDK,INRGDK,ISOLDK,IONDK,                                    MSM02060
     1     IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,   MSM02070
     2     THRMAT,CHID,ID,RID,SCTFRC,IFLAV,BNDMAX,ALPHA,                MSM02080
     3     BETA,TETA,PHI,BIR,SINI,TIMTAG,COLAT,                         MSM02090
     4     ALOCT,V,VNORTH,VSOUTH,ASOUTH,BSOUTH,DXS,DYS,                 MSM02100
     5     ANORTH,BNORTH,DXN,DYN,VM,VMLOSS,BMIN,XMIN,YMIN,ZMIN,         MSM02110
     6     ALAM,ETABEG,ETABND,FLXBND,BNDLOC,ETA,FLUX,EFLUX,FLXSUM,EAVG, MSM02120
     7     R,P,                                                         MSM02130
     8     ALMDEL,NAUGEL,AUGPAR,MODE,RPP,THRSH,ERSHFT,DKTIME)           MSM02140
C                                                                       MSM02150
C  VERSION 1.1                   DATE: NOVEMBER 23, 1988                MSM02160
C          1.2                         SEPTEMBER 8, 1989                MSM02170
C          1.3                         NOVEMBER 17, 1989                MSM02180
C          1.4 (APRIL, 1990 VERSION)      MARCH 26, 1990                MSM02190
C          1.5 (JULY 1, 1990 VERSION)      JULY 01, 1990                MSM02200
C                                                                       MSM02210
C  PROGRAMMER:  BRYAN BALES/R.W. SPIRO                                  MSM02220
C                                                                       MSM02230
C  PURPOSE:  THIS EXECUTIVE ROUTINE CONTROLS THE ELECTRIC AND MAGNETIC  MSM02240
C          FIELD MODEL (EBCON) AND THE RUNNING OF THE PARTICLE          MSM02250
C          TRACE ALGORITHM (PPTM).                                      MSM02260
C                                                                       MSM02270
C  CALLING PARAMETERS:                                                  MSM02280
C     ISTART       START YEAR, DAY, AND TIME (SEC)                      MSM02290
C     IINC         MAJOR TIME STEP INCREMENT (YEAR, DAY, TIME)          MSM02300
C     IEND         ENDING YEAR, DAY, AND TIME (SEC)                     MSM02310
C     LATDIM       NUMBER OF LATITUDINAL GRID SPACES                    MSM02320
C     LTDIM        NUMBER OF LOCAL TIME (LONGITUDINAL) GRID SPACES      MSM02330
C     IEDIM        NUMBER OF PARTICLE SPECIES TO BE TRACED              MSM02340
C     ITMDIM       MAXIMUM NUMBER OF MAJOR TIME STEPS FOR THIS RUN      MSM02350
C     IRDK         R DIMENSION OF DKTIME ARRAY                          MSM02360
C     INRGDK       ENERGY DIMENSION OF DKTIME ARRAY                     MSM02370
C     ISOLDK       SUN SPOT NUMBER DIMENSION OF DKTIME ARRAY            MSM02380
C     IONDK        NUMBER OF ION SPECIES IN DKTIME ARRAY                MSM02390
C     IRDIM        R DIMENSION OF FLXMAT ARRAYS                         MSM02400
C     NRGDIM       ENERGY DIMENSION OF FLXMAT ARRAYS                    MSM02410
C     KPDIM        NUMBER OF KP VALUES FOR FLXMAT ARRAYS                MSM02420
C     KPPLUS       AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)           MSM02430
C     ISPDIM       PLASMA MASS SPECIES DIMENSION                        MSM02440
C     FLXR         R VALUES AT WHICH FLXMAT ARRAYS ARE CALCULATED       MSM02450
C     FLXNRG       ENERGY VALUES FOR WHICH FLXMAT ARRAYS ARE CALCULATED MSM02460
C     FLXKP        KP VALUES FOR WHICH FLXMAT ARRAYS ARE CALCULATED     MSM02470
C     FLXMAT       ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,  MSM02480
C                  AND KP FOR USE IN CALCULATING INITIAL AND BOUNDARY   MSM02490
C                  FLUX VALUES.(COMPUTED IN CALL TO FLXNIT)             MSM02500
C     THRMAT       ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,  MSM02510
C                  AND KP FOR USE AS THRESHOLD FLUXES (COMPUTED IN CALL MSM02520
C                  TO FLXNIT)                                           MSM02530
C     CHID         CHARACTER HEADER STRING (UP TO 80 CHARACTERS)        MSM02540
C     ID           INTEGER HEADER VECTOR                                MSM02550
C                  ID(1) = YEAR                                         MSM02560
```

```
C                ID(2)  = DAY
C                ID(3)  = HOURS
C                ID(4)  = MINUTES
C                ID(5)  = SECONDS
C                ID(6)  = PRESENTLY UNUSED
C                ID(7)  = PRESENTLY UNUSED
C                ID(8)  = FIRST DIMENSION OF OUTPUT ARRAY
C                ID(9)  = SECOND DIMENSION OF OUTPUT ARRAY
C                ID(10) = THIRD DIMENSION OF OUTPUT ARRAY
C                ID(11) = TIME INDEX L
C                ID(12) - ID(20) = PRESENTLY UNUSED
C     RID        REAL HEADER VECTOR
C                RID(1) = TIMTAG
C                RID(2) = KP
C                RID(3) = POLAR CAP POTENTIAL DROP
C                RID(4) = TIME DERIVATIVE OF LOCATION OF
C                         LOW-LATITUDE EDGE OF AURORAL PRECIPITATION
C                RID(5) - RID(20) = PRESENTLY UNUSED
C     SCTFRC     ASSUMED PITCH ANGLE SCATTERING EFFICIENCY
C     IFLAV      CHEMICAL SPECIES IDENTIFIER
C                1 MEANS ELECTRONS
C                2 MEANS H+ IONS
C                3 MEANS O+ IONS
C     BNDMAX     MAXIMUM I VALUE OF MODEL BOUNDARY ALONG EACH LOCAL
C                TIME GRID LINE DURING THE DURATION OF THIS RUN.
C     ALPHA      LATITUDINAL GRID SPACING VECTOR
C     BETA       LONGITUDINAL GRID SPACING VECTOR
C     TETA       COLATITUDE OF I GRID LINES (RADIANS) (USED ONLY IN
C                SUBROUTINE RDGRID)
C     PHI        NON-ROTATED HOUR ANGLE OF J GRID LINES (USED ONLY
C                IN SUBROUTINE RDGRID)
C     BIR        RADIAL COMPONENT OF IONOSPHERIC MAGNETIC FIELD (NT)
C     SINI       SIN(MAGNETIC FIELD INCLINATION ANGLE)
C     TIMTAG     VECTOR GIVING TIMES AT WHICH E AND B PARAMETERS ARE
C                CALCULATED
C     COLAT      GRID COLATITUDE ARRAY (RADIANS)
C     ALOCT      GRID LOCAL TIME ARRAY (RADIANS EASTWARD FROM NOON)
C     V          ELECTRIC POTENTIAL DISTRIBUTION ON GRID (OUTPUT FROM
C                EBCON) (AVERAGE OF VNORTH AND VSOUTH)
C     VNORTH     NORTHERN HEMISPHERE ELECTRIC POTENTIAL DISTRIBUTION
C                (OUTPUT FROM EBCON)
C     VSOUTH     SOUTHERN HEMISPHERE ELECTRIC POTENTIAL DISTRIBUTION
C                (OUTPUT FROM EBCON)
C     ASOUTH, BSOUTH, DSX, DYS, ANORTH, BNORTH, DXN, DYN
C                SPECIFICATIONS OF ELECTRIC FIELD MODEL BOUNDAY
C                ELLIPSES
C     VM         FLUX TUBE VOLUME (OUTPUT FROM EBCON)
C     VMLOSS     GEOMETRIC FACTOR USED IN CALCULATING PRECIPITATION
C                LOSS
C     BMIN       EQUATORIAL MAGNETIC FIELD STRENGTH (NT)
C     XMIN       GSM X LOCATION OF WHERE FIELD LINE GOING THROUGH
C                GRID PT CROSSES THE EQUATORIAL (B-FIELD MIMIMUM)
C                PLANE
C     YMIN       GSM Y LOCATION OF WHERE FIELD LINE GOING THROUGH
C                GRID PT CROSSES THE EQUATORIAL PLANE
C     ZMIN       GSM Z LOCATION OF WHERE FIELD LINE GOING THROUGH
C                GRID PT CROSSES THE (B-FIELD MINIMUM) PLANE
C     ALAM       ENERGY INVARIANT VECTOR
C     ETABEG     INITIAL FLUX TUBE CONTENT DISTRIBUTION EACH TIME
C                STEP
C     ETABND     BOUNDARY FLUX TUBE CONTENT DISTRIBUTION
C     FLXBND     FLUX VALUES AT BOUNDARY
C     BNDLOC     LOCATION OF OUTER BOUNDARY OF DETAILED PARTICLE
C                TRACES
```

```
C     ETA          FLUX TUBE CONTENT ARRAY (PARTICLES/UNIT MAGNETIC     MSM03210
C                                                      FLUX)            MSM03220
C     FLUX         PARTICLE FLUX AT GRID PTS (PARTICLES/CM**2/S)        MSM03230
C     EFLUX        PRECIPITATING ENERGY FLUX ARRAY                      MSM03240
C     FLXSUM       TOTAL PRECIPITATION ENERGY FLUX ARRAY (INTEGRATED    MSM03250
C                  OVER SPECIES) FOR (1) ELECTRONS (2) H+ & (3) O+      MSM03260
C     EAVG         AVERAGE PRECIPITATING ELECTRON ENERGY                MSM03270
C                  FOR (1) ELECTRONS (2) H+ AND (3) O+                  MSM03280
C     R            ARRAY GIVING RADIAL DISTANCE OF GRID PTS IN          MSM03290
C                  MAGNETOSPHERIC EQUATORIAL PLANE (RE)                 MSM03300
C     P            ARRAY GIVING HOUR ANGLE (MEASURED EASTWARD FROM      MSM03310
C                  NOON IN RADIANS) OF GRID PTS IN EQUATORIAL PLANE     MSM03320
C     ALMDEL       WIDTH OF ENERGY INVARIANT CHANNELS                   MSM03330
C     NAUGEL       DIMENSION OF INPUT ARRAYS                            MSM03340
C     AUGPAR       AUGMENTED DATA ARRAY FOR INPUT VALUES                MSM03350
C     MODE         LOGICAL VARIABLE TELLING WHICH INPUT PARAMETERS WE   MSM03360
C                  HAVE VALUES FOR                                      MSM03370
C     RPP          NOMINAL RADIUS OF PLASMAPAUSE DEFINED AS DIPOLE      MSM03380
C                  MAPPING OF EQUATORWARD EDGE OF AURORAL ZONE TO       MSM03390
C                  MAGNETOSPHERIC EQUATORIAL PLANE                      MSM03400
C     DKTIME       ARRAY GIVING CHARGE EXCHANGE DECAY TIMES (SECONDS)   MSM03410
C                  FOR AN ARRAY OF RADIAL DISTANCES, ENERGIES, SUN SPOT MSM03420
C                  NUMBERS, AND SPECIES                                 MSM03430
C                                                                       MSM03440
C                                                                       MSM03450
C                                                                       MSM03460
      CHARACTER*80 CHID                                                 MSM03470
C                                                                       MSM03480
      INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,         MSM03490
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,           MSM03500
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,          MSM03510
     3   STAND,PCP,DEQDT,IPATT                                          MSM03520
C                                                                       MSM03530
      COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,       MSM03540
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,           MSM03550
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,          MSM03560
     3   STAND,PCP,IPATT,DDST,DEQDT                                     MSM03570
C                                                                       MSM03580
C                                                                       MSM03590
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,           MSM03600
     1   LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,          MSM03610
     2   LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,       MSM03620
     3   LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH            MSM03630
      COMMON /MOVE/IBEG,ISTOP,JBEG,JSTOP,IEBEG,IEEND                    MSM03640
      COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG                              MSM03650
C                                                                       MSM03660
      DIMENSION ISTART(3),IINC(3),IEND(3),ITIME(3),                     MSM03670
     1 ETA(LATDIM,LTDIM,IEDIM),EFLUX(LATDIM,LTDIM,IEDIM),               MSM03680
     2 FLXSUM(LATDIM,LTDIM,3),EAVG(LATDIM,LTDIM,3),                     MSM03690
     3 FLUX(LATDIM,LTDIM,IEDIM)                                         MSM03700
C                                                                       MSM03710
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),V(LATDIM,LTDIM,ITMDIM),         MSM03720
     1 ETABEG(LATDIM,LTDIM,IEDIM),ETABND(LTDIM,IEDIM,ITMDIM),           MSM03730
     2 FLXBND(LTDIM,IEDIM,ITMDIM),                                      MSM03740
     3 ALAM(IEDIM),COLAT(LATDIM,LTDIM),TIMTAG(ITMDIM),                  MSM03750
     4 ALOCT(LATDIM,LTDIM),BMIN(LATDIM,LTDIM),XMIN(LATDIM,LTDIM),       MSM03760
     5 YMIN(LATDIM,LTDIM),ZMIN(LATDIM,LTDIM),BNDLOC(LTDIM,ITMDIM),      MSM03770
     6 ALMDEL(IEDIM)                                                    MSM03780
C                                                                       MSM03790
     DIMENSION IFLAV(IEDIM),BNDMAX(LTDIM),                              MSM03800
     1 ALPHA(LATDIM),BETA(LATDIM),BIR(LATDIM,LTDIM),                    MSM03810
     2 ASOUTH(3,ITMDIM),BSOUTH(3,ITMDIM),DXS(3,ITMDIM),DYS(3,ITMDIM),   MSM03820
     3 ANORTH(3,ITMDIM),BNORTH(3,ITMDIM),DXN(3,ITMDIM),DYN(3,ITMDIM),   MSM03830
     4 VMLOSS(LATDIM,LTDIM,ITMDIM),                                     MSM03840
```

```
      5   TETA(LATDIM),PHI(LTDIM),SINI(LATDIM,LTDIM)                    MSM03850
C                                                                       MSM03860
          DIMENSION VNORTH(LATDIM,LTDIM),VSOUTH(LATDIM,LTDIM)           MSM03870
C                                                                       MSM03880
          DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)       MSM03890
C                                                                       MSM03900
          DIMENSION SCTFRC(IEDIM,LTDIM,ITMDIM),RPP(ITMDIM)              MSM03910
C                                                                       MSM03920
          DIMENSION AUGPAR(NAUGEL,ITMDIM)                               MSM03930
          LOGICAL*1 MODE(NAUGEL,ITMDIM)                                 MSM03940
C                                                                       MSM03950
C                                                                       MSM03960
          DIMENSION IDGHV(2),ID(20),RID(20),THRSH(IEDIM)                MSM03970
          DIMENSION ERSHFT(IEDIM)                                       MSM03980
          DIMENSION ITMNEW(3),IRESTR(3)                                 MSM03990
C                                                                       MSM04000
          DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)                  MSM04010
          DIMENSION THRMAT(IRDIM,NRGDIM,KPDIM)                          MSM04020
          DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)             MSM04030
C                                                                       MSM04040
          DIMENSION DKTIME(IRDK,INRGDK,ISOLDK,IONDK)                    MSM04050
C                                                                       MSM04060
          DATA TSHENG,TSHING /100.0,50./                                MSM04070
C                                                                       MSM04080
C                                                                       MSM04090
C      SET UP THE GRID                                                  MSM04100
C                                                                       MSM04110
          CALL RDGRID(LATDIM,LTDIM,ALPHA,BETA,TETA,PHI,COLAT,BIR,SINI)  MSM04120
C                                                                       MSM04130
C       ***** WRITE ALL VALUES COMPUTED IN RDGRID                       MSM04140
C                                                                       MSM04150
          WRITE (6,*) 'LATDIM = ',LATDIM                                MSM04160
          WRITE (6,*) 'LTDIM = ',LTDIM                                  MSM04170
          WRITE (6,*) 'ALPHA ', ALPHA                                   MSM04180
          WRITE (6,*) 'BETA  ', BETA                                    MSM04190
          WRITE (6,*) 'TETA  ', TETA                                    MSM04200
          WRITE (6,*) 'PHI   ', PHI                                     MSM04210
          CALL OUTP(COLAT,LATDIM,LTDIM,1,LATDIM,1,3,LTDIM,3,0.,         MSM04220
     1     'COLAT',6,132)                                               MSM04230
C         CALL OUTP(BIR,LATDIM,LTDIM,10,LATDIM,1,1,LTDIM,1,0.,'BIR',6,132MSM04240
C    1     )                                                            MSM04250
C         CALL OUTP(SINI,LATDIM,LTDIM,10,LATDIM,2,3,LTDIM,3,0.,         MSM04260
C    1     'SINI',6,80)                                                 MSM04270
C                                                                       MSM04280
C                                                                       MSM04290
C  READ TABLE OF CHARGE EXCHANGE DECAY TIMES FOR USE IN COMPUTING       MSM04300
C      ION CHARGE EXCHANGE LOSS                                         MSM04310
C                                                                       MSM04320
          CALL READDK(IRDK,INRGDK,ISOLDK,IONDK,DKTIME)                  MSM04330
C                                                                       MSM04340
C  WRITE OUT DECAY TIMES                                                MSM04350
          DO 4 IR=1,IRDK                                                MSM04360
             ELVAL=1.+.5*FLOAT(IR)                                      MSM04370
             WRITE(6,865) ELVAL                                         MSM04380
  865 FORMAT(1X/1X,'L-VALUE = ',F5.2)                                   MSM04390
             DO 6 ION=1,2                                               MSM04400
                IF(ION.EQ.1) WRITE(6,866)                               MSM04410
                IF(ION.EQ.2) WRITE(6,867)                               MSM04420
  866 FORMAT(1X,'H+ DECAY TIMES: (SECONDS)')                            MSM04430
  867 FORMAT(1X,'O+ DECAY TIMES: (SECONDS)')                            MSM04440
                WRITE(6,868) ((DKTIME(IR,INRG,ISOL,ION),INRG=1,INRGDK), MSM04450
     2                        ISOL=1,ISOLDK)                            MSM04460
  868 FORMAT(13(1X,1PE9.3))                                             MSM04470
    6        CONTINUE                                                   MSM04480
```

```
  4     CONTINUE                                                        MSM04490
C                                                                       MSM04500
C                                                                       MSM04510
C  SET UP TIME INDEPENDENT ENERGY CHANNELS TO MODEL                     MSM04520
C                                                                       MSM04530
        CALL SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)                      MSM04540
        IEEND=IEMAX                                                     MSM04550
C                                                                       MSM04560
C  WRITE OUT ENERGY CHANNEL INFORMATION                                 MSM04570
        DO 999 IE=1,IEMAX                                               MSM04580
           WRITE(6,*) 'IFLAV=',IFLAV(IE),'  ALAM=',ALAM(IE),            MSM04590
     2                 '  ALMDEL=',ALMDEL(IE)                           MSM04600
         THRSH(IE)=0.0                                                  MSM04610
999     CONTINUE                                                        MSM04620
C                                                                       MSM04630
C  INITIALIZE MATRIX OF KP DEPENDENT DEFAULT FLUXES                     MSM04640
C                                                                       MSM04650
      CALL FLXNIT(IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,   MSM04660
     2            FLXMAT,THRMAT)                                        MSM04670
C                                                                       MSM04680
C                                                                       MSM04690
C   SPECIAL PRINTOUT FOR FLXMAT AND THRMAT                              MSM04700
      DO 300 IR=1,IRDIM                                                 MSM04710
         IF(IR.EQ.1) RDIST=3.                                           MSM04720
         IF(IR.EQ.2) RDIST=4.                                           MSM04730
         IF(IR.EQ.3) RDIST=6.6                                          MSM04740
         IF(IR.EQ.4) RDIST=13.                                          MSM04750
C                                                                       MSM04760
        DO 310 IKP=1,KPPLUS                                             MSM04770
           MYKP=IKP-1                                                   MSM04780
C          WRITE(6,850) RDIST,MYKP                                      MSM04790
  850 FORMAT(1X,'R=',F6.2,2X,'KP=',I2,/)                                MSM04800
C          WRITE(6,*) 'ELECTRONS'                                       MSM04810
C          WRITE(6,851)                                                 MSM04820
  851 FORMAT(1X,'LOG ENERGY',5X,'LOG FLUX',5X,'LOG THRESHOLD')          MSM04830
C                                                                       MSM04840
C                                                                       MSM04850
           DO 320 INRG=1,NRGDIM                                         MSM04860
C             WRITE(6,852) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,1),          MSM04870
C    2                     THRMAT(IR,INRG,IKP)                          MSM04880
  852 FORMAT(F10.4,5X,F8.4,5X,F13.4)                                    MSM04890
C                                                                       MSM04900
  320      CONTINUE                                                     MSM04910
C                                                                       MSM04920
C          WRITE(6,853)                                                 MSM04930
  853 FORMAT(1X/)                                                       MSM04940
C                                                                       MSM04950
C          WRITE(6,*) 'H+ IONS'                                         MSM04960
           DO 321 INRG=1,NRGDIM                                         MSM04970
C             WRITE(6,859) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,2)           MSM04980
  859 FORMAT(F10.4,5X,F8.4)                                             MSM04990
C                                                                       MSM05000
  321      CONTINUE                                                     MSM05010
C                                                                       MSM05020
C          WRITE(6,853)                                                 MSM05030
C                                                                       MSM05040
C          WRITE(6,*) 'O+ IONS'                                         MSM05050
           DO 322 INRG=1,NRGDIM                                         MSM05060
C             WRITE(6,859) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,3)           MSM05070
C                                                                       MSM05080
  322      CONTINUE                                                     MSM05090
C                                                                       MSM05100
C          WRITE(6,853)                                                 MSM05110
  310   CONTINUE                                                        MSM05120
```

```
  300 CONTINUE
C
C
      WRITE(6,*) 'ELECTRONS'
      WRITE(6,*) 'FLXMAT ARRAYS (IR,NRG,IKP); IR=1 IS 3RE, IR=2 IS 4RE'
      WRITE(6,*) 'IR=3 IS 6.6 RE, IR=4 IS 13 RE; NRG CORRESPONDS TO'
      WRITE(6,*) 'LOG OF ENERGY IN EV RUNNING FROM 10 EV TO 10**8 EV'
      WRITE(6,*) 'ENERGY= 10.**((3+NRG)/4); KP=IKP-1'
      WRITE(6,*) 'KP=0'
      CALL OUTP(FLXMAT(1,1,1,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=1'
      CALL OUTP(FLXMAT(1,1,2,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=2'
       CALL OUTP(FLXMAT(1,1,3,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=3'
      CALL OUTP(FLXMAT(1,1,4,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=4'
      CALL OUTP(FLXMAT(1,1,5,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=5'
      CALL OUTP(FLXMAT(1,1,6,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'KP=6'
      CALL OUTP(FLXMAT(1,1,7,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'MINIMA'
      CALL OUTP(FLXMAT(1,1,8,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'MAXIMA'
      CALL OUTP(FLXMAT(1,1,9,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT1',6,132)
      WRITE(6,*) 'H+ IONS'
      WRITE(6,*) 'KP=0'
      CALL OUTP(FLXMAT(1,1,1,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=1'
      CALL OUTP(FLXMAT(1,1,2,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=2'
       CALL OUTP(FLXMAT(1,1,3,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=3'
      CALL OUTP(FLXMAT(1,1,4,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=4'
      CALL OUTP(FLXMAT(1,1,5,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=5'
      CALL OUTP(FLXMAT(1,1,6,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'KP=6'
      CALL OUTP(FLXMAT(1,1,7,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'MINIMA'
      CALL OUTP(FLXMAT(1,1,8,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'MAXIMA'
      CALL OUTP(FLXMAT(1,1,9,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT2',6,132)
      WRITE(6,*) 'O+ IONS'
```

```
      WRITE(6,*) 'KP=0'
      CALL OUTP(FLXMAT(1,1,1,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=1'
      CALL OUTP(FLXMAT(1,1,2,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=2'
       CALL OUTP(FLXMAT(1,1,3,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=3'
      CALL OUTP(FLXMAT(1,1,4,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=4'
      CALL OUTP(FLXMAT(1,1,5,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=5'
      CALL OUTP(FLXMAT(1,1,6,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'KP=6'
      CALL OUTP(FLXMAT(1,1,7,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'MINIMA'
      CALL OUTP(FLXMAT(1,1,8,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      WRITE(6,*) 'MAXIMA'
      CALL OUTP(FLXMAT(1,1,9,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'FLXMAT3',6,132)
      CALL OUTP(THRMAT(1,1,1),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT1',6,80)
      CALL OUTP(THRMAT(1,1,2),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT2',6,80)
      CALL OUTP(THRMAT(1,1,3),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT3',6,80)
      CALL OUTP(THRMAT(1,1,4),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT4',6,80)
      CALL OUTP(THRMAT(1,1,5),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT5',6,80)
      CALL OUTP(THRMAT(1,1,6),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT6',6,80)
      CALL OUTP(THRMAT(1,1,7),IRDIM,NRGDIM,1,IRDIM,1,1,NRGDIM,1,0.,
     2          'THRMAT7',6,80)
C
C     GET CURRENT TIME
C
c         CALL DTIME(ISTEB)
C
C
          IRECST=MAX(1,IRECEB)
C
C  CALL TO EBCON COMPUTES ALL THE ELECTRIC AND MAGNETIC FIELD MODELS
C     NEEDED FOR THE RUN
C
          CALL EBCON(ISTART,IINC,IEND,COLAT,ALOCT,LATDIM,LTDIM,ITMDIM,
     1             IEDIM,NAUGEL,CHID,ID,RID,VM,BMIN,XMIN,YMIN,ZMIN,
     2             V,VNORTH,VSOUTH,ASOUTH,BSOUTH,DXS,DYS,
     3             ANORTH,BNORTH,DXN,DYN,BNDLOC,
     4             TIMTAG,ITMMAX,R,P,BIR,SINI,VMLOSS,SCTFRC,AUGPAR,MODE,
     5             RPP,IEMAX,TETA)
C
C     GET CURRENT TIME AND FIND TIME SPENT IN EBCON
C
c         CALL DTIME(IENDEB)
c         WRITE (LUERR,*) 'THE TIME SPENT IN EBCON IS ',
c    1                  ABS(IENDEB-ISTEB)/100.,' SECONDS.'
```

```
C
C       COMPUTE BNDMAX FROM BNDLOC
C
        DO 8 LT=1,LTDIM
           BNDMAX(LT)=0.
           DO 9 ITM=1,ITMMAX
              IF(BNDLOC(LT,ITM).GT.BNDMAX(LT))
     1           BNDMAX(LT)=BNDLOC(LT,ITM)
 9         CONTINUE
 8      CONTINUE
C
C
C
C
C
C
C
C      FIND THE TIME DEPENDENDENT PLASMA BOUNDARY CONDITION FOR PPTM
C
        CALL BNDSET(LATDIM,LTDIM,IEDIM,ITMDIM,ITMMAX,
     1      IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
     2      NAUGEL,TIMTAG,BNDLOC,IFLAV,AUGPAR,R,P,VM,ETA,ETABND,
     3      FLXBND,ALAM,ALMDEL,IEMAX)
C
C
        CALL OUTP(FLXBND(1,1,1),LTDIM,IEDIM,3,LTDIM,1,1,IEMAX,1,
     2         0.,'FLXBND(J,IE,1)',6,132)
C
C
C
      IF(IRECPT.EQ.0)THEN
        AZERO=.2
        BZERO=0.
        CZERO=0.
      ELSE
C
C  READ START FLUX FROM DATA SET
        IRECMX=ABS(IRECPT)
        IRDFLX=(IRECMX-1)*IEMAX+1
        CALL RDHDR(LUFLX,'AFLX  ',IRDFLX,ID,RID,CHID,
     2              LATDIM,LTDIM,ITMDIM)
C
C       CHECK IF TIME IS CORRECT
C
        ISEC=ID(3)*3600+ID(4)*60+ID(5)
        IF(ID(1).NE.ISTART(1).OR.
     1     ID(2).NE.ISTART(2).OR.
     2     ISEC.NE.ISTART(3)) THEN
C
           WRITE(6,*) 'START TIME DOES NOT MATCH ETABEG START TIME'
           WRITE(6,*) 'STOPPING PROGRAM IN PPTCON'
           WRITE(6,*) 'ISTART=',ISTART
           WRITE(6,*) 'ISEC=',ISEC,'IRECMX=',IRECMX
C
           STOP
        END IF
C
        CALL READ3D(LUFLX,'AFLX  ',IRECMX,LATDIM,LTDIM,IEMAX,ITMDIM,
     1           ID,RID,CHID,FLUX)
C
C  CONVERT FROM PARTICLE FLUX TO INVARIANT FLUX ETA
C
        CALL FLX2ET(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,1,IFLAV,ALAM,
     2             ALMDEL,BNDLOC,VM,FLUX,ETA)
```

```
      C
      C
      C
              AZERO=0.
              BZERO=.2
              CZERO=0.
      C
            END IF
      C
              IRECPT=MAX(1,IRECPT)
              LL=1
      C
      C
      C  INITIALIZE PLASMA DISTRIBUTION
      C
              CALL INITAL(LL,LATDIM,LTDIM,IEDIM,ITMDIM,IRDIM,
           2           NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,
           3       FLXMAT,IFLAV,AUGPAR(KP,LL),R,P,VM,ETA,ETABEG,ALAM,ALMDEL,
           4       BNDLOC,AZERO,BZERO,CZERO,IEMAX)
      C
      C  CONVERT ETABEG TO FLUX AND WRITE OUT TO DISK FILE
              CALL ET2FLX(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,1,IFLAV,ALAM,
           2              ALMDEL,BNDLOC,VM,FLUX,ETABEG)
      C
              ID(1)=ISTART(1)
              ID(2)=ISTART(2)
              ID(3)=ISTART(3)/3600
              ID(4)=MOD(ISTART(3),3600)/60
              ID(5)=MOD(ISTART(3),60)
              ID(11)=1
              ID(8)=LATDIM
              ID(9)=LTDIM
              ID(10)=IEMAX
      C
              CALL WRT3D(LUFLX,IRECPT,ID,RID,CHID,FLUX,LATDIM,LTDIM,IEMAX,
           2             ITMDIM)
      C
      C
              DO 400 KK=1,IEMAX
                WRITE(6,*) 'KK= ',KK
                CALL OUTP(ETABEG(1,1,KK),LATDIM,LTDIM,20,45,1,3,LTDIM-1,
           2         1,0.,'ETABEG(KK)',6,132)
                WRITE(6,*) 'KK=',KK,'BEGINNING FLUX'
                CALL OUTP(FLUX(1,1,KK),LATDIM,LTDIM,20,45,1,3,LTDIM-1,
           2         1,0.,'FLUX(KK)',6,132)
        400   CONTINUE
      C
      C
              ITIME(1) = ISTART(1)
              ITIME(2) = ISTART(2)
              ITIME(3) = ISTART(3)
      C
      C
      C
      C
      C
      C  FOLLOWING IS THE MAIN TIME LOOP IN THE PROGRAM
      C  LOOP OVER ALL TIMES IN TIMTAG (L=1 CORRESPONDS TO INITIAL TIME)
            DO 10 L=2,ITMMAX
              IRECPT=IRECPT+1
              TSTART=TIMTAG(L)
              TSTOP=TIMTAG(L-1)
              PSTEP=TSTART-TSTOP
              WRITE(6,*) ' L= ',L,' IRECPT= ',IRECPT,'TSTART= ',TSTART
```

```
          WRITE(6,*) ' TSTOP= ',TSTOP,' PSTEP= ',PSTEP,'ITMMAX=',ITMMAX
C
          CALL OUTP(ETABND(1,1,L),LTDIM,IEDIM,3,LTDIM,1,1,IEMAX,1,
     2         0.,'ETABND(J,IE,L)',6,132)
C
C
             DTNOM=-600.
             IF(ABS(DTNOM).GT.PSTEP) DTNOM=-PSTEP
C
             LL=L
C
C
C
C
             IMODE=0
C
            IF(MODE(KP,L).AND.IMODE.EQ.0) THEN
C                                     PERFORM FULL PARTICLE TRACES
C
C     GET THE CURRENT TIME
c            CALL DTIME(ISTPPT)
C
             CALL PPTM(TSTART,TSTOP,DTNOM,SCTFRC,IFLAV,BNDMAX,ALPHA,BETA,
     1        BIR,ITMMAX,LATDIM,LTDIM,IEDIM,ITMDIM,IRDIM,NRGDIM,KPDIM,
     2        KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,THRMAT,TIMTAG,ALAM,
     2        ALMDEL,VM,R,P,VMLOSS,V,ETABEG,ETABND,BNDLOC,ETA,EFLUX,
     3        FLXSUM,EAVG,LL,RPP,AUGPAR(KP,LL),IEMAX,THRSH,TSHENG,
     4        TSHING,IRDK,INRGDK,ISOLDK,IONDK,DKTIME)
C
C     GET THE CURRENT TIME AND FIND THE TIME SPENT IN PPTM
C
c            CALL DTIME(IENPPT)
c            WRITE(LUERR,*)' THE TIME SPENT IN PPTM IS ',
c    1                  ABS(IENPPT-ISTPPT)/100.,' SECONDS.'
C
C   CALL PWRCAL TO ESTIMATE ELECTRON PRECIPITATION POLEWARD OF MAIN
C   MSM MODELING REGION
C
             CALL PWRCAL(LATDIM,LTDIM,3,ITMDIM,LL,AUGPAR(KP,LL),FLXSUM,
     2                  EAVG,COLAT,ALOCT,ALPHA,BETA,BNDLOC,ANORTH,
     3                  BNORTH,DXN,DYN)
C
               WRITE(18,*)' TIME = ',TIMTAG(LL)
               WRITE(6,*)' TIME = ',TIMTAG(LL)
C
            ELSE
C
C  USE EMPIRICAL FLXMAT DENSITIES
C
c             CALL DTIME(ISTFLX)
C
              CALL FLXDFL(LL,AUGPAR(KP,LL),R,P,ALAM,VM,IRDIM,NRGDIM,
     1          KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,
     2          ALMDEL,LATDIM,LTDIM,IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,
     3          1,IEMAX)
C
              DO 998 IDO=1,IEMAX
               IF(IFLAV(IDO).EQ.1) THEN
                 THRSH(IDO)=TSHENG
               ELSE
                 THRSH(IDO)=TSHING
               END IF
998           CONTINUE
C
```

```
C
c           CALL DTIME(INDFLX)
c           WRITE(LUERR,*)' THE TIME SPEND IN FLXDFL IS ',
c    1        ABS(INDFLX-ISTFLX)/100.,' SECONDS.'
c           CALL DTIME(ISTAUR)
          ENDIF
C
C
C  CALL TO AURL2S PRODUCES HARDY ION PRECIPITATION MODEL
            CALL AURL2S(AUGPAR(KP,L),COLAT,ALOCT,LATDIM,LTDIM,
     1         FLXSUM(1,1,2),EAVG(1,1,2))
C           CALL OUTP(FLXSUM(1,1,2),LATDIM,LTDIM,15,LATDIM,1,3,
C    1         LTDIM,1,0.,'IFLXSUM',6,132)
C
c          CALL DTIME(IENAUR)
c          WRITE(LUERR,*)' THE TIME SPENT IN AURL  IS ',
c    1                  ABS(IENAUR-ISTAUR)/100.,' SECONDS.'
C
C
      WRITE(6,*) 'T=',TSTART
C
C     CONVERT TSTOP TO 3 VALUED TIME AND PUT IT INTO ITIME
C     GET ALOCT AT TSTOP
C
           ITIME(3) = ITIME(3) + INT(PSTEP)
 11        CONTINUE
              IF (ITIME(3).GT.86400) THEN
                 ITIME(2) = ITIME(2) + 1
                 ITIME(3) = ITIME(3) - 86400
                 IF (MOD(ITIME(1),4).NE.0.AND.ITIME(2).GT.365) THEN
                    ITIME(2) = ITIME(2) - 365
                    ITIME(1) = ITIME(1) + 1
                 ELSEIF (MOD(ITIME(1),4).EQ.0.AND.ITIME(2).GT.366) THEN
                    ITIME(2) = ITIME(2) - 366
                    ITIME(1) = ITIME(1) + 1
                 END IF
                 GO TO 11
              END IF
C
C     ---->   WRITE FLUX, EFLUX, FLXSUM, EAVG
C     ---->          TO ARCHIVAL FILES
C
           DO 2 IND=1,20
              ID(IND)=0
              RID(IND)=0.
    2      CONTINUE
      WRITE(6,*) 'L=',L,'ITIME=',ITIME(3)
         ID(1)=ITIME(1)
         ID(2)=ITIME(2)
         ID(3)=ITIME(3)/3600
         ID(4)=MOD(ITIME(3),3600)/60
         ID(5)=MOD(ITIME(3),60)
         ID(11)=L
C
C     ----> WRITE FLXBND TO ARCHIVE FILE
C
          ID(8)=LTDIM
          ID(9)=IEDIM
          ID(10)=1
C
      CALL WRT3D(LUFLXB,IRECPT,ID,RID,CHID,FLXBND(1,1,L),LTDIM,IEDIM,
     2         1,ITMDIM)
C
C
```

```
      ID(8)=LATDIM
      ID(9)=LTDIM
      ID(10)=1
C
C
C    ----> WRITE EAVG AND FLXSUM TO ARCHIVE FILES
C
      CALL WRT3D(LUEAVG,IRECPT,ID,RID,CHID,EAVG,LATDIM,LTDIM,1,ITMDIM)
      CALL WRT3D(LUFLSM,IRECPT,ID,RID,CHID,FLXSUM,LATDIM,LTDIM,1,ITMDIM)
      CALL WRT3D(LUPIFX,IRECPT,ID,RID,CHID,FLXSUM(1,1,2),LATDIM,LTDIM,
     2          1,ITMDIM)
      CALL WRT3D(LUPIEN,IRECPT,ID,RID,CHID,EAVG(1,1,2),LATDIM,LTDIM,
     2          1,ITMDIM)
C
C    ----> WRITE FLUX AND EFLUX TO ARCHIVE FILES
C
      ID(8)=LATDIM
      ID(9)=LTDIM
      ID(10)=IEMAX
C
C  CONVERT ETA TO FLUX FOR OUTPUT
C
         CALL ET2FLX(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,L,IFLAV,ALAM,
     2               ALMDEL,BNDLOC,VM,FLUX,ETA)
C
C
      CALL WRT3D(LUFLX,IRECPT,ID,RID,CHID,FLUX,LATDIM,LTDIM,IEMAX,
     1   ITMDIM)
C     CALL WRT3D(LUEFLX,IRECPT,ID,RID,CHID,EFLUX,LATDIM,LTDIM,IEMAX,
C    1               ITMDIM)
C
      IUT=100*ID(3)+ID(4)
      DO 111 IE=1,IEMAX
      WRITE(6,*) 'IE=',IE,' ALAM(IE)=',ALAM(IE),' SPECIES=',IFLAV(IE),
     1          'L=',L,' DAY=',ID(2),' ITIME=',ITIME(3),'UT=',IUT
C        CALL OUTP(ETA(1,1,IE),LATDIM,LTDIM,15,45,1,3,LTDIM,1,0.,
C    1    'ETA',6,132)
         CALL OUTP(FLUX(1,1,IE),LATDIM,LTDIM,15,45,1,3,LTDIM,1,0.,
     1    'FLUX',6,132)
  111 CONTINUE
C
            CALL OUTP(ALOCT,LATDIM,LTDIM,10,11,1,3,LTDIM,1,0.,
     1        'ALOCT',6,132)
         WRITE(6,*) 'L=',L,'DAY=',ID(2),'UT=',IUT,'ITIME=',ITIME(3)
         CALL OUTP(FLXSUM,LATDIM,LTDIM,15,55,1,3,LTDIM,1,0.,
     1    'FLXSUM',6,132)
         WRITE(6,*) 'L=',L,'DAY=',ID(2),'UT=',IUT,'ITIME=',ITIME(3)
         CALL OUTP(EAVG,LATDIM,LTDIM,15,55,1,3,LTDIM,3,0.,
     1    'EAVG',6,132)
C
C     COPY CURRENT VALUES IN ETA TO ETABEG
C
            DO 20, I=1, LATDIM
               DO 30, J=1, LTDIM
                  DO 40, K=1, IEMAX
                     ETABEG(I,J,K) = ETA(I,J,K)
 40               CONTINUE
 30            CONTINUE
 20         CONTINUE
C
   10 CONTINUE
      CALL OPTERR(ETA,VM,LATDIM,LTDIM,IEDIM,IEMAX,ALAM,ALMDEL,THRSH,
     1  ITMDIM,IFLAV,ISTART,IEND,R,P,TIMTAG,ITMMAX,ERSHFT)
C
```

```fortran
         RETURN
      END
C
C ***************************** E B C O N ****************************
C
      SUBROUTINE EBCON(ISTART,IINC,IEND,COLAT,ALOCT,LATDIM,LTDIM,ITMDIM,
     1         IEDIM,NAUGEL,CHID,ID,RID,VM,BMIN,XMIN,YMIN,ZMIN,V,VNORTH,
     2         VSOUTH,ASOUTH,BSOUTH,DXS,DYS,ANORTH,BNORTH,DXN,DYN,BNDLOC,
     3         TIMTAG,ITMMAX,R,P,BIR,SINI,VMLOSS,SCTFRC,AUGPAR,MODE,RPP,
     4         IEMAX,TETA)
C
C  VERSION 1.1                     DATE: FEBRUARY 5, 1988
C                                         09.12.89
C
C  PROGRAMMER: BRYAN A. BALES/R.W. SPIRO
C
C  PURPOSE:  EBCON CALLS ROUTINES TO CALCULATE THE ELECTRIC AND THE
C         MAGNETIC FIELD AT OUR GRID POINTS.
C
C
         CHARACTER*80 CHID
C
C
         INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,DEQDT,IPATT
C
         COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,IPATT,DDST,DEQDT
C
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
         COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG
C
         DIMENSION AUGPAR(NAUGEL,ITMDIM)
         DIMENSION ISTART(3),IINC(3),IEND(3),ITIME(3),
     1    ITMNEW(3),VM(LATDIM,LTDIM,ITMDIM),V(LATDIM,LTDIM,ITMDIM),
     2    BMIN(LATDIM,LTDIM),XMIN(LATDIM,LTDIM),TIMTAG(ITMDIM),
     3    YMIN(LATDIM,LTDIM),ZMIN(LATDIM,LTDIM),
     4    COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM)
C
         DIMENSION VNORTH(LATDIM,LTDIM), VSOUTH(LATDIM,LTDIM)
         DIMENSION TETA(LATDIM)
C
        DIMENSION BNDLOC(LTDIM,ITMDIM),
     2   ASOUTH(3,ITMDIM),BSOUTH(3,ITMDIM),DXS(3,ITMDIM),DYS(3,ITMDIM),
     3   ANORTH(3,ITMDIM),BNORTH(3,ITMDIM),DXN(3,ITMDIM),DYN(3,ITMDIM),
     4   BIR(LATDIM,LTDIM),SINI(LATDIM,LTDIM),
     5   VMLOSS(LATDIM,LTDIM,ITMDIM),SCTFRC(IEDIM,LTDIM,ITMDIM),
     6   RPP(ITMDIM)
C
         DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)
C
         LOGICAL*1 MODE(NAUGEL,ITMDIM)
C
         DIMENSION ID(20),RID(20)
C
         ITIME(IYEAR) = ISTART(IYEAR)
```

```
        ITIME(IDAY) = ISTART(IDAY)
        ITIME(ISECND) = ISTART(ISECND)
C
C
      PI=ATAN2(0.,-1.)
C
C  INITIALIZE ID ARRAYS
        DO 4 IC=1,20
          ID(IC)=0
          RID(IC)=0.
    4   CONTINUE
C
C  FECON RETURNS TIME-DEPENDENT INPUT PARAMETERS
C
c         CALL DTIME(ISTFE)
C
          CALL FECON(ISTART,IINC,IEND,ITMDIM,NAUGEL,MODE,AUGPAR,
     2                 ITMMAX,TIMTAG)
C
          WRITE (6,*) 'ITMMAX = ',ITMMAX
          DO 101, I=1, ITMMAX
            WRITE (6,800) I,TIMTAG(I)
  101     CONTINUE
  800 FORMAT (' TIMTAG(',I2.2,') = ',F15.1)
C
C     GET THE CURRENT TIME AND FIND THE TIME SPENT IN FECON
C
c         CALL DTIME(IENDFE)
c         WRITE (LUERR,*) 'THE TIME SPENT IN FECON IS ',
c    1                    ABS(IENDFE-ISTFE)/100.,' SECONDS.'
C
C
C  BEGINNING OF EB TIME LOOP
        IRECEB=MAX(1,IRECEB)
        IRECEB=IRECEB-1
C
      DO 10 IT=1,ITMMAX
          L = IT
          IRECEB=IRECEB+1
C
C     SET ALOCT TO CORRECT VALUE
C
          CALL MLTSET(ITIME(ISECND),LATDIM,LTDIM,ALOCT)
C
C  CALCULATE SCTFRC ARRAY = FRACTION OF STRONG PITCH ANGLE SCATTERING
C
C
          FKP=AUGPAR(KP,L)
          CALL SETSCT(L,ITIME(ISECND),IEDIM,LTDIM,ITMDIM,LATDIM,
     2                  FKP,ALAM,ALOCT,SCTFRC,IEMAX)
C
C
C      ***** PRINT OUT ALOCT
C
          WRITE (6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
     1      MOD(ITIME(ISECND),3600)/60
          CALL OUTP(ALOCT,LATDIM,LTDIM,10,11,1,3,LTDIM,1,0.,
     1     'ALOCT',6,132)
C
C     GET THE CURRENT TIME
C
C
C      ***** WRITE AUGPAR
C
```

```
          WRITE (6,*) 'AUGPAR PARAMETERS'
          WRITE(6,810)
  810 FORMAT(1X,'   IYEAR',2X,'    IDAY',2X,'   IHOUR',2X,'    IMIN',
     2       2X,'    ISEC',2X,'      KP',2X,'     DST',2X,'  EQEDGE',
     3       2X,'  DLATAZ',2X,'  DLATRV',2X,'   MLTRV')
          WRITE(6,820) (AUGPAR(LM,IT),LM=1,11)
  820 FORMAT(1X,11(F8.2,2X))
          WRITE(6,830)
  830 FORMAT(1X)
          WRITE(6,840)
  840 FORMAT(1X,'   IMFBX',2X,'   IMFBY',2X,'   IMFBZ',2X,'  CLAPSE',
     2       2X,'  GEOMGX',2X,'  GEOMGY',2X,'  GEOMGZ',2X,'  GEOMGT',
     3       2X,'  LBOUND',2X,'   SWVEL',2X,'   SWDEN')
          WRITE(6,820) (AUGPAR(LM,IT),LM=12,22)
          WRITE(6,830)
          WRITE(6,850)
  850 FORMAT(1X,'   TILTW',2X,'   STAND',2X,'     PCP',2X,'   IPATT',
     2       2X,'    DDST',2X,'   DEQDT')
          WRITE(6,820) (AUGPAR(LM,IT),LM=23,28)
          WRITE(6,830)
C
C
C
C
C     GET THE CURRENT TIME
C
c         CALL DTIME(ISTBT)
C
C  CALCULATE APPROXIMATE PLASMAPAUSE RADIUS FOR USE IN PPTM FOR
C    DETERMINING WEAK LOSS RATE
      RPP(L)=1./(COS(AUGPAR(EQEDGE,L)*PI/180.))**2
C
C
C  CALCULATE B FIELD MATRICES
C
          FSTOFF=AUGPAR(STAND,L)
          FEQEDG=AUGPAR(EQEDGE,L)
          FDST=AUGPAR(DST,L)
          FCLPSE=AUGPAR(CLAPSE,L)
          FTILT=AUGPAR(TILTW,L)
C
          WRITE (6,*) 'L =',L,' FSTOFF =',FSTOFF,' FEQEDG =',FEQEDG,
     1                ' FDST =',FDST,' FCLPSE =',FCLPSE,' FTILT =',
     2                FTILT,'  RPP =',RPP(L)
C
          CALL BTRACE(LATDIM,LTDIM,ITMDIM,L,FSTOFF,FEQEDG,FDST,
     2         FCLPSE,FTILT,VM,BMIN,XMIN,YMIN,ZMIN,R,P,ALOCT,TETA)
C
C     GET THE CURRENT TIME AND FIND THE TIME SPENT IN BTRACE
C
c         CALL DTIME(IENDBT)
c         WRITE (LUERR,*) 'THE TIME SPENT IN BTRACE IS ',
c    1                ABS(IENDBT-ISTBT)/100.,' SECONDS.'
C
      WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
     1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
      WRITE(6,*) 'L=',L
C
          ICNTRL = 1
C
C     GET THE CURRENT TIME
C
c         CALL DTIME(ISTEF)
C
```

```
C
      WRITE(6,*) 'IN EBCON'
      WRITE(6,*) 'EQEDGE= ',AUGPAR(EQEDGE,L),' STAND= ',AUGPAR(STAND,L)
C GET BNDYS FOR USE IN EFIELD
      VDROP=AUGPAR(PCP,L)*1.E3
      CALL EFBNDY(LATDIM,LTDIM,ITMDIM,L,AUGPAR(EQEDGE,L),
     2            AUGPAR(STAND,L),VDROP,R,ALOCT,COLAT,
     3            ASOUTH(1,L),BSOUTH(1,L),DXS(1,L),DYS(1,L))
      CALL EFBNDY(LATDIM,LTDIM,ITMDIM,L,AUGPAR(EQEDGE,L),
     2            AUGPAR(STAND,L),VDROP,R,ALOCT,COLAT,
     3            ANORTH(1,L),BNORTH(1,L),DXN(1,L),DYN(1,L))
      WRITE(6,*) 'AFTER EFBNDY L= ',L,' A= ',(ANORTH(IIC,L),IIC=1,3)
      WRITE(6,*) 'BNORTH=',(BNORTH(IIC,L),IIC=1,3)
      WRITE(6,*) '   DXN= ',(DXN(IIC,L),IIC=1,3)
      WRITE(6,*) '   DYN= ',(DYN(IIC,L),IIC=1,3)
C
C
C
            IPATRN=AUGPAR(IPATT,L)
            CALL EFIELD(IPATRN,VDROP,AUGPAR(DEQDT,L),
     2         ASOUTH,BSOUTH,DXS,DYS,ANORTH,BNORTH,DXN,DYN,COLAT,
     3         ALOCT,MODE,LATDIM,LTDIM,JWRAP,ITMDIM,L,ICNTRL,1,V,
     4         VNORTH,VSOUTH)
C
C     GET THE CURRENT TIME AND FIND THE TIME SPENT IN EFIELD
C
c           CALL DTIME(IENDEF)
c           WRITE (LUERR,*) 'THE TIME SPENT IN EFIELD IS ',
c    1                   ABS(IENDEF-ISTEF)/100.,' SECONDS.'
C
C      ***** WRITE V
C
            CALL OUTP(V(1,1,L),LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
     1         'V',6,132)
      WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
     1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
      WRITE(6,*) 'L=',L
C
C           CALL OUTP(VNORTH,LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
C    2         'VNORTH',6,132)
C           CALL OUTP(VSOUTH,LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
C    2         'VSOUTH',6,132)
C
C
C
C     FIND BNDLOC FOR THIS TIME
C
            CALL FNDBND(L,COLAT,ALOCT,ANORTH(2,L),BNORTH(2,L),
     1         DXN(2,L),DYN(2,L),LATDIM,LTDIM,ITMDIM,BNDLOC)
C
C COMPUTE RBNDY
            WRITE(6,*) 'L= ',L
            DO 8 J=1,LTDIM
               BI=BNDLOC(J,L)
               BJ=J
               BL=L
               RBNDY=G3NTRP(R,LATDIM,LTDIM,ITMDIM,BI,BJ,BL)
               WRITE(6,*) 'J=',J,' BNDLOC=',BNDLOC(J,L),' RBNDY=',RBNDY
    8       CONTINUE
C
C
C
            IBM=LATDIM
            DO 6 JJ=1,LTDIM
```

```
             IF(INT(BNDLOC(JJ,L)).LT.IBM) THEN
                IBM=INT(BNDLOC(JJ,L))
             END IF
    6     CONTINUE
C
C
C       ***** WRITE VM, BMIN, XMIN, YMIN, ZMIN
C
      WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
     1           MOD(ITIME(ISECND),3600)/60
      WRITE(6,*) 'L=',L
          CALL OUTP(VM(1,1,L),LATDIM,LTDIM,IBM,LATDIM,1,3,LTDIM-1,1,
     1       1.,'VM',6,132)
C         CALL OUTP(BMIN,LATDIM,LTDIM,IBM,LATDIM,2,3,LTDIM-3,3,10.,
C    1       'BMIN',6,132)
C     WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
C    1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
C     WRITE(6,*) 'L=',L
C         CALL OUTP(XMIN,LATDIM,LTDIM,IBM,LATDIM,2,3,LTDIM-3,3,1.,
C    1       'XMIN',6,132)
C     WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
C    1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
C     WRITE(6,*) 'L=',L
C         CALL OUTP(YMIN,LATDIM,LTDIM,IBM,LATDIM,2,3,LTDIM-3,3,1.,
C    1       'YMIN',6,132)
C     WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
C    1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
C     WRITE(6,*) 'L=',L
C         CALL OUTP(ZMIN,LATDIM,LTDIM,IBM,LATDIM,2,3,LTDIM-3,3,1.,
C    1       'ZMIN',6,132)
      WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
     1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
      WRITE(6,*) 'L=',L
          CALL OUTP(R(1,1,L),LATDIM,LTDIM,IBM,45,1,1,LTDIM,1,1.,
     1       'R',6,132)
C
C     WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
C    1           MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
C     WRITE(6,*) 'L=',L
C         CALL OUTP(P(1,1,L),LATDIM,LTDIM,IBM,LATDIM,1,1,LTDIM,1,0.01,
C    1       'P',6,132)
C     SET VMLOSS FOR THE CURRENT VM PARAMETERS
C
          CALL VMLSET(L,VM,BIR,SINI,LATDIM,LTDIM,ITMDIM,VMLOSS)
C
C       ***** WRITE VMLOSS
C
C         CALL OUTP(VMLOSS(1,1,L),LATDIM,LTDIM,IBM,LATDIM,2,3,LTDIM-3,
C    1       3,0.01,'VMLOSS',6,132)
C
C  SET UP ID ARRAYS
C
      WRITE(6,*) 'BEFORE ID ARRAY LOADED L=',L
          ID(1)=ITIME(IYEAR)
          ID(2)=ITIME(IDAY)
          ID(3)=ITIME(ISECND)/3600
          ID(4)=MOD(ITIME(ISECND),3600)/60
          ID(5)=MOD(ITIME(ISECND),60)
          ID(11)=L
          RID(1)=TIMTAG(L)
          RID(2)=AUGPAR(KP,L)
          RID(3)=AUGPAR(PCP,L)
          RID(4)=AUGPAR(DEQDT,L)
C
```

```
C
C     -----> WRITE AUGPAR TO AN ARCHIVAL FILE ******
            ID(8)=NAUGEL
            ID(9)=1
            ID(10)=ITMMAX
            CALL WRT3D(LUPDAT,IRECEB,ID,RID,CHID,AUGPAR(1,L),NAUGEL,
     2                 1,1,ITMDIM)
C
C
C     -----> WRITE VM, BMIN, XMIN, YMIN, ZMIN, AND V TO AN ARCHIVAL FILE
C
            ID(8)=LATDIM
            ID(9)=LTDIM
            ID(10)=1
C
            CALL WRT3D(LUBFLD,IRECEB,ID,RID,CHID,VM(1,1,L),
     1                 LATDIM,LTDIM,1,ITMDIM)
            CALL WRT3D(LUBMIN,IRECEB,ID,RID,CHID,BMIN,
     1                 LATDIM,LTDIM,1,ITMDIM)
            CALL WRT3D(LUXMIN,IRECEB,ID,RID,CHID,XMIN,LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUYMIN,IRECEB,ID,RID,CHID,YMIN,LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUZMIN,IRECEB,ID,RID,CHID,ZMIN,LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUEFLD,IRECEB,ID,RID,CHID,V(1,1,L),LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUVN,IRECEB,ID,RID,CHID,VNORTH,LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUVS,IRECEB,ID,RID,CHID,VSOUTH,LATDIM,
     1                 LTDIM,1,ITMDIM)
C
C    ----> WRITE ALOCT, COLAT, AND BNDLOC TO ARCHIVAL FILE
C
            ID(8)=LATDIM
            ID(9)=LTDIM
            ID(10)=1
C
            CALL WRT3D(LUALOC,IRECEB,ID,RID,CHID,ALOCT,LATDIM,
     1                 LTDIM,1,ITMDIM)
            CALL WRT3D(LUCOLT,IRECEB,ID,RID,CHID,COLAT,LATDIM,
     1           LTDIM,1,ITMDIM)
C
            ID(8)=LTDIM
            ID(9)=1
            ID(10)=1
C
            CALL WRT3D(LUBNDL,IRECEB,ID,RID,CHID,BNDLOC(1,L),
     1                  LTDIM,1,1,ITMDIM)
C
C
C
C     INCREMENT ITIME
C
   12 CONTINUE
C
      CALL TIMINC(ITIME,ITMNEW,IINC)
C
      ITIME(IYEAR)=ITMNEW(IYEAR)
      ITIME(IDAY)=ITMNEW(IDAY)
      ITIME(ISECND)=ITMNEW(ISECND)
C
C
C
```

```
   10 CONTINUE
C
C
C
         RETURN
      END
C
C
C ************************ B L O C K   D A T A ************************
C
      BLOCK DATA
C
C  VERSION 1.0                    DATE: JANUARY 9, 1988
C
C  PROGRAMMER: BRYAN A. BALES
C
C  PURPOSE:  THIS BLOCK DATA CONTAINS THE INDEX OF DATA ELEMENTS IN THE
C         PARRAY AND AUGPAR ARRAYS.
C
C
         INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,DEQDT,IPATT
C
         COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,IPATT,DDST,DEQDT
C
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
         COMMON /GRID/DLAM,DPSI,RI,JWRAP
C
         COMMON /MOVE/IBEG,ISTOP,JBEG,JSTOP,IEBEG,IEEND
C
         DATA IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,IPATT,DDST,DEQDT
     4      /1,2,3,3,4,5,6,7,
     5       8,9,10,11,12,13,14,15,
     6       16,17,18,19,20,21,22,23,
     7       24,25,26,27,28/
C
         DATA LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
     4       /9,13,14,15,16,17,
     5        20,21,22,23,24,25,26,27,
     6        28,35,36,37,38,39,40,41,
     7        42,43,44,45,46,47,48,49/
      DATA IBEG,ISTOP,JBEG,JSTOP,IEBEG /1,53,3,50,1/
C
      END
C
      SUBROUTINE EMODEL(IPATT,PCP,DEQDT,A,B,DX,DY,COLAT,ALOCT,MODE,
     1 LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,V)
C
C     THIS SUBROUTINE RETURNS VALUES OF THE POTENTIAL V AT ALL GRID
```

```
C     POINTS, SPECIFIED BY COLAT(I,J) AND ALOCT(I,J).                   MSM14090
C                                                                       MSM14100
C                                                                       MSM14110
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-    MSM14120
C              87-K-0001.                                               MSM14130
C                                                                       MSM14140
C                                                                       MSM14150
C                                                                       MSM14160
C    GENERAL CALL PARAMETERS FOR SUBROUTINE EMODEL                      MSM14170
C                                                                       MSM14180
C       IPATT = HEPPNER-MAYNARD PATTERN NO.  MUST BE SPECIFIED AS AN    MSM14190
C               INTEGER BETWEEN 1 AND 7.                                MSM14200
C               IPATT=1 MEANS PATTERN A, FOR BZ SOUTH, BY=0.            MSM14210
C               IPATT=2 MEANS PATTERN BC, FOR BZ SOUTH, BYNORTH>0       MSM14220
C               IPATT=3 MEANS PATTERN DE, FOR BZ SOUTH,BYNORTH<0        MSM14230
C               IPATT=4 MEANS PATTERN BCP,TWISTED BC, FOR BZ>0          MSM14240
C               IPATT=5 MEANS PATTERN BCPP,TWISTED BC,BZ STRONG>0       MSM14250
C               IPATT=6 MEANS PATT. DEP, TWISTED DE,BZ WEAK >0          MSM14260
C               IPATT=7 MEANS PATT. DEPP, TWISTED DE,BZ STRONG>0        MSM14270
C                                                                       MSM14280
C       PCP = POLAR-CAP POTENTIAL DROP IN VOLTS.  MUST BE SPECIFIED     MSM14290
C             UNLESS USING UNSCALED HEPPNER-MAYNARD POTENTIAL(ICNTRL=-1)MSM14300
C                                                                       MSM14310
C       DEQDT = ESTIMATED D/DT OF LATITUDE OF EQUATORWARD EDGE OF       MSM14320
C               AURORAL ZONE AT LOCAL MIDNIGHT, IN DEGREES/HOUR.  THIS  MSM14330
C               IS A PARAMETER IN LOW-LATITUDE E-FIELD MODEL.  NOT      MSM14340
C               NEEDED IF ICNTRL.LE.0.                                  MSM14350
C                                                                       MSM14360
C      VECTORS A(3), B(3), DX(3), DY(3) DESCRIBE THE ELLIPSES THAT FORM MSM14370
C      THE BOUNDARIES BETWEEN REGIONS 1, 2, AND 3.  BOUNDARY 1 IS THE   MSM14380
C      EQUATORWARD EDGE OF THE POLAR CAP.  BOUNDARY 2 IS THE EQUATORWARDMSM14390
C      EDGE OF REGION 1 OR THE EQUATORWARD EDGE OF THE FIELD-REVERSAL   MSM14400
C      REGION.  BOUNDARY 3 IS THE SHIELDING LAYER.                      MSM14410
C       A(L) = RADIUS OF ELLIPSE MEASURED IN X(SUNWARD) DIRECTION.      MSM14420
C       B(L) = RADIUS OF ELLIPSE MEASURED IN Y(DUSKWARD) DIRECTION.     MSM14430
C       DX(L) = SUNWARD DISPLACEMENT OF COORD.SYSTEM CENTER FROM POLE.  MSM14440
C       DY(L) = DUSKWARD DISPL. OF COORD. SYSTEM CENTER FROM POLE.      MSM14450
C      TO USE EFIELD WITH RCM, CHOOSE A(2) = B(2) = TETA(IMIN),         MSM14460
C      DX(2) = -OFFSET*180./PI, DY(2) = 0.  ELLIPSE PARAMETERS FOR      MSM14470
C      BOUNDARIES 1 AND 3 NEED NOT BE SPECIFIED.                        MSM14480
C                                                                       MSM14490
C      COLAT AND ALOCT ARE THE USUAL MAGNETIC COLATITUDE AND MAGNETIC   MSM14500
C      LOCAL TIME ANGLES IN RADIANS.                                    MSM14510
C                                                                       MSM14520
C      MODE IS A DUMMY PARAMETER, NOT CURRENTLY USED.                   MSM14530
C                                                                       MSM14540
C      IMAX, JMAX, JWRAP HAVE THEIR USUAL MEANING, SAME AS IN RCM.      MSM14550
C                                                                       MSM14560
C      ITMDIM = NUMBER OF TIME LABELS IN THE V-MATRIX.  FOR USE WITH    MSM14570
C               RCM, IT SHOULD BE SET EQUAL TO 1.                       MSM14580
C                                                                       MSM14590
C      ITMCUR = TIME-LABEL NUMBER OF V-MATRIX CURRENTLY BEING COMPUTED. MSM14600
C               FOR USE WITH RCM, IT SHOULD BE SET EQUAL TO 1.          MSM14610
C                                                                       MSM14620
C      ICNTRL = 1 MEANS DIFFERENT FORMULAS ARE USED IN REGIONS 1,2,3.   MSM14630
C             = 0 MEANS HEPPNER-MAYNARD FORMULA IS USED FOR ALL         MSM14640
C                 LATITUDES, BUT SCALED TO EXTERNALLY SPECIFIED PCP AND MSM14650
C                 ELLIPSE PARAMETERS.                                   MSM14660
C             = -1 MEANS HEPPNER-MAYNARD FORMULA IS USED FOR ALL        MSM14670
C                  LATITUDES, UNSCALED.                                 MSM14680
C             = -2 IS THE MODE USED FOR USE WITH RCM.  HEPPNER-MAYNARD  MSM14690
C                  IS USED POLEWARD OF ELLIPSE 2, AND ON ELLIPSE 2,     MSM14700
C                  SCALED TO FIT SPECIFIED POTENTIAL DROP AND DIMENSIONSMSM14710
C                  OF ELLIPSE 2.                                        MSM14720
```

```
C
C      ITOP = 1 MEANS THAT A(1), B(1), DX(1), DY(1) ARE COMPUTED
C               INTERNALLY IN EFIELD.  OTHERWISE THEY MUST BE PASSED TO
C               EFIELD FROM OUTSIDE.  ITOP IS NORMALLY = 1 FOR USE IN
C
C               MSM.  ITOP SHOULD BE SET TO 1 FOR USE WITH RCM.
C      V = POTENTIAL MATRIX COMPUTED IN EMODEL.  IT IS DIMENSIONED AT
C          LATDIM X LTDIM
C
C
C
C
C     SUBROUTINE EMODEL CALLS SUBROUTINES EPOT(HEPPNER-MAYNARD),
C     LOW(LOW-LATITUDES), AURORA(REGION 2), REG1(REGION 1).
C
C
C
      PARAMETER(JDIM=51)
      DIMENSION THETAA(JDIM),THETAB(JDIM),THETAC(JDIM)
      DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM)
      DIMENSION V(LATDIM,LTDIM)
      DIMENSION VA(JDIM),DVADT(JDIM),VC(JDIM),DVCDT(JDIM)
      DIMENSION VB(JDIM),VBLOW(JDIM),VBAZ(JDIM),DVBLO(JDIM)
      DIMENSION DVBDT(JDIM),DVBAZ(JDIM)
      DIMENSION COEF(18,18),MLBL(5)
      DIMENSION A(3), B(3), DX(3), DY(3)
      DIMENSION XCO(18,18),AHM(2,7),BHM(2,7),DXHM(2,7)
      DIMENSION DYHM(2,7),VMIN(7),VMAX(7)
C
C     PARAMETER VALUES
C
C       NTAPE = FILE NUMBER FROM WHICH MAYNARD-RICH COEFFICIENTS ARE
C               READ.
C
      IMAX = LATDIM
      JMAX = LTDIM
      IF(JDIM.NE.LTDIM) THEN
         WRITE(6,184) JDIM, LTDIM
  184    FORMAT(1H ,'JDIM = ',I4,'  LTDIM = ',I4,'IN EFIELD.')
      END IF
C
      WRITE(6,*) 'IPATT,PCP,DEQDT',IPATT,PCP,DEQDT
C
      PI = 3.141593
      PI2 = 2.*PI
      NTAPE = 7
      NNMAX = 16
      EPS = .01
      G = 1.
      IEFF = IMAX
      CALL INPUT(NTAPE,COEF,IPATT,XCO,AHM,BHM,DXHM,DYHM,VMIN,
     1    VMAX)
C
C   FG = SCALING FACTOR THAT CONVERTS HM POTENTIAL TO PRESENT SITUATION
C
      IF(ICNTRL.GE.0) THEN
         FG = PCP/(VMAX(IPATT)-VMIN(IPATT))
      ELSE
         FG = 1.
      END IF
      WRITE(6,903) IPATT,VMAX(IPATT),VMIN(IPATT),FG
 903  FORMAT(1H ,'IPATT,VMAX,VMIN,FG = ',I5,3E15.5)
C
C     COMPUTE A(1),B(1),DX(1),DY(1) IF ITOP = 1.
```

```
C
       IF(ITOP.EQ.1) THEN
          A(1) = A(2)*AHM(1,IPATT)/AHM(2,IPATT)
          B(1) = B(2)*BHM(1,IPATT)/BHM(2,IPATT)
         DX(1) = DX(2) + A(2)*(DXHM(1,IPATT)-DXHM(2,IPATT))/AHM(2,IPATT)
         DY(1) = DY(2) + B(2)*(DYHM(1,IPATT)-DYHM(2,IPATT))/BHM(2,IPATT)
      END IF
C
      VMAXX = VMAX(IPATT)*FG
      VMINN = VMIN(IPATT)*FG
      SA=SIN((A(3)-DX(3))*PI/180.)
      VPENET = -13112.*G*DEQDT*SA
      VBAR = 0.6*VMAXX + 0.4*VMINN -0.22*VPENET
C
      IF(ICNTRL.NE.-2) THEN
         WRITE(6,905) IPATT,VMAXX,VMINN
 905     FORMAT(1H ,'IPATT,VMAXX,VMINN = ',I5,2E15.5)
         WRITE(6,906) A(1),A(2),A(3)
 906     FORMAT(1H ,'A(1), A(2), A(3) = ',3E15.5)
         WRITE(6,907) B(1),B(2),B(3)
 907     FORMAT(1H ,'B(1), B(2), B(3) = ',3E15.5)
         WRITE(6,908) DX(1), DX(2), DX(3)
908      FORMAT(1H ,'DX(1), DX(2), DX(3) = ',3E15.5)
         WRITE(6,909) DY(1), DY(2), DY(3)
 909     FORMAT(1H ,'DY(1), DY(2), DY(3) = ',3E15.5)
         WRITE(6,899) VPENET,VBAR,VMAXX,VMINN
 899     FORMAT(1H ,'VPENET,VBAR,VMAX,VMIN=',4E15.5)
C
      END IF
C
C
      IF(ICNTRL.EQ.0.OR.ICNTRL.EQ.-1) GO TO 15
C
C    COMPUTATION OF BOUNDARY LOCATIONS AS FCNS OF J.
C
C   GUARD AGAINST IRRELEVANT A'S OR B'S BEING ZERO OR NEGATIVE.
C
      DO 811 L=1,3
         IF(A(L).LT.0.001) A(L)=0.001
         IF(B(L).LT.0.001) B(L) = 0.001
 811  CONTINUE
C
C
      DO 1 J = 1, JMAX
         AL = ALOCT(IEFF,J)
         THETAA(J) = THET(A(1),B(1),DX(1),DY(1),AL)*PI/180.
         THETAB(J) = THET(A(2),B(2),DX(2),DY(2),AL)*PI/180.
         THETAC(J) = THET(A(3),B(3),DX(3),DY(3),AL)*PI/180.
 1    CONTINUE
C
C   SEGMENT FOR DEFINING BOUNDARY POTENTIALS AND E-FIELDS
C
      DO 7 J=1, JMAX
C
C     POTENTIAL AND DERIVATIVE AT BOUNDARY A
C
         GLONG = ALOCT(IEFF,J) + PI
         IF (GLONG.GT.(2.*PI)) GLONG = GLONG - 2.*PI
         GMLAT = 90. - THETAA(J)*180./PI
         CALL EPOT(COEF,GMLAT,GLONG,VA(J),NTAPE,IPATT,NNMAX,A,B,
     1   DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
         GMLAT = GMLAT - EPS*180./PI
         CALL EPOT(COEF,GMLAT,GLONG,VT,NTAPE,IPATT,NNMAX,A,B,
     1   DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
```

```
         DVADT(J) = (VT-VA(J))/EPS
    7  CONTINUE
C
C      POTENTIAL AND DERIVATIVE AT C
C
       DO 8 J=1, JMAX
         CALL LOW(THETAC(J),ALOCT(IEFF,J),VC(J),DEQDT,A(3),DX(3),VBAR,G)
         TT = THETAC(J) + EPS
         CALL LOW(TT, ALOCT(IEFF,J),VT,DEQDT,A(3),DX(3),VBAR,G)
         DVCDT(J) = (VT-VC(J))/EPS
    8  CONTINUE
C
C
C      POTENTIAL AT B
C
       DO 9 J=1,JMAX
         GLONG = ALOCT(IEFF,J)+PI
         IF(GLONG.GT.(2.*PI)) GLONG = GLONG-2.*PI
         GMLAT = 90. - THETAB(J)*180./PI
         CALL EPOT(COEF,GMLAT,GLONG,VB(J),NTAPE,IPATT,NNMAX,A,B,
     1     DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
         VBLOW(J) = VC(J) - DVCDT(J)*(THETAC(J)-THETAB(J))
         VBAZ(J) = VB(J) - VBLOW(J)
    9  CONTINUE
C
C
C    DERIVATIVE AT POTENTIAL AT B
C
       DO 10 J = 1, JMAX
         DVBLO(J) = DVCDT(J)
         TT = THETAB(J) + EPS
         CALL AURORA(TT,ALOCT(IEFF,J),VT,VBAZ,THETAB(J),THETAC(J),J,
     1    JMAX,JWRAP)
         DVBAZ(J) = (VT-VBAZ(J))/EPS
          DVBDT(J) = DVBLO(J) + DVBAZ(J)
   10  CONTINUE
       NMAX=NNMAX
C
C      MAIN DO LOOP OVER GRID
C
   15  CONTINUE
       DO 5 I = 1, IMAX
       DO 5 J = 1, JMAX
         V(I,J)= 0.
C
C
C
C      CALLS OF DIFFERENT SUBROUTINE SEQUENCES
C
C      REGION ZERO  =  POLAR CAP
C
         IF(COLAT(I,J).LE.THETAA(J).OR.ICNTRL.EQ.0.OR.ICNTRL.EQ.-1) THEN
           GLONG = ALOCT(I,J) + PI
           IF (GLONG.GT.PI2) GLONG=GLONG-PI2
           GMLAT = 90. - COLAT(I,J)*180./PI
           CALL EPOT(COEF,GMLAT,GLONG,VTEMP,NTAPE,IPATT,NNMAX,A,
     1       B,DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
         END IF
C
C     REGION 3 = LOW LATITUDES
C
       IF(COLAT(I,J).GE.THETAC(J).AND.ICNTRL.GT.0) THEN
         CALL LOW(COLAT(I,J),ALOCT(I,J),VTEMP,DEQDT,A(3),DX(3),VBAR,G)
C
```

```
      END IF
C
C
C      REGION 2 = LOWER AURORAL ZONE
C
      IF(COLAT(I,J).LT.THETAC(J).AND.COLAT(I,J).GE.THETAB(J).AND.ICNTRL.
     1 GT.0) THEN
         VLOW = VC(J) + DVCDT(J) * (COLAT(I,J)-THETAC(J))
         CALL AURORA(COLAT(I,J),ALOCT(I,J),VAZ,VBAZ,THETAB(J),THETAC(J),
     %    J, JMAX,JWRAP)
         VTEMP = VAZ + VLOW
      END IF
C
      IF(COLAT(I,J).LT.THETAB(J).AND.COLAT(I,J).GT.THETAA(J).AND.ICNTRL.
     1  GT.0) THEN
         CALL REG1(COLAT(I,J),ALOCT(I,J),VTEMP,THETAA(J),
     %    THETAB(J),VA(J),VB(J),DVADT(J),DVBDT(J))
      END IF
C
C    USE HEPPNER-MAYNARD DIRECTLY IN REGION 1 FOR USE WITH RCM.
C
      TB = THETAB(J) + 1.E-4
      IF(COLAT(I,J).LT.TB.AND.COLAT(I,J).GT.THETAA(J).AND.ICNTRL.EQ.-2)
     1 THEN
         GLONG = ALOCT(I,J) + PI
         IF(GLONG.GT.PI2) GLONG = GLONG - PI2
         CALL EPOT(COEF,GMLAT,GLONG,VTEMP,NTAPE,IPATT,NNMAX,A,
     1      B,DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
      END IF
C
      V(I,J)= VTEMP
C
    5 CONTINUE
      RETURN
      END
C
C
C
C
C
C
      SUBROUTINE EPOT(COEF,TLAT,TLON,VALUE,NTAPE,IPATT,NNMAX,A,B,
     1   DX, DY, XCO, AHM, BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
C
C  PURPOSE:  THIS SUBROUTINE USES THE SCALED HEPPNER-MAYNARD MODEL TO
C            RETURN THE ELECTRIC POTENTIAL AT A SPECIFIC POINT.
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C
C  INPUTS:
C    COEF          COEFFICIENTS FROM DIGITIZATION OF HEPPNER-MAYNARD
C                  E-FIELD MODEL
C    TLAT          COLATITUDE OF PT WHERE V IS TO BE CALCULATED
C                  (RADIANS)
C    TLON          MLT OF PT WHERE V IS TO BE CALCULATED
C    NTAPE         UNIT NUMBER OF FILE FROM WHICH COEFFICIENTS ARE READ
C    IPATT         HEPPNER-MAYNARD PATTERN TYPE
C    NNMAX         NUMBER OF COEFFICIENTS
C    A             RADIUS OF ELLIPSE IN X (SUNWARD) DIRECTION
C    B             RADIUS OF ELLIPSE IN Y (DUSKWARD) DIRECTION
C    DX            OFFSET OF ELLIPSE IN X DIRECTION
C    DY            OFFSET OF ELLIPSE IN Y DIRECTION
```

```
C    XCO          COEFFICIENTS                                          MSM17290
C    AHM          HEPPNER-MAYNARD X RADIUS                              MSM17300
C    BHM          HEPPNER-MAYNARD Y RADIUS                              MSM17310
C    DXHM         HEPPNER-MAYNARD X OFFSET                              MSM17320
C    DYHM         HEPPNER-MAYNARD Y OFFSET                              MSM17330
C    VMIN         ELECTRIC POTENTIAL MIN FROM HEPPNER-MAYNARD MODEL     MSM17340
C    VMAX         ELECTRIC POTENTIAL MAX FROM HEPPNER-MAYNARD MODEL     MSM17350
C    PCP          MEASURED CROSS-POLAR-CAP POTENTIAL DROP               MSM17360
C    ICNTRL       = 1 MEANS DIFFERENT FORMULAS USED IN REGIONS 1,2,3    MSM17370
C                 = 2 H-M USED AT ALL LATITUDES, BUT SCALED TO PCP AND MSM17380
C                   ELLIPSE PARAMETERS                                  MSM17390
C                 =-1 H-M USED AT ALL LATITUDES, UNSCALED               MSM17400
C                 =-2 SHOULD NOT BE USED. RESERVED FOR USE WITH RICE    MSM17410
C                   CONVECTION MODEL                                    MSM17420
C                                                                       MSM17430
C  OUTPUT:                                                              MSM17440
C    VALUE        ELECTRIC POTENTIAL (VOLTS)                            MSM17450
                                                                        MSM17460
C                                                                       MSM17470
      DIMENSION AHM(2,7),BHM(2,7),DXHM(2,7),DYHM(2,7)                   MSM17480
      DIMENSION VMIN(7),VMAX(7)                                         MSM17490
      DIMENSION A(3), B(3), DX(3),DY(3)                                 MSM17500
      DIMENSION COEF(18,18),P(18,18)                                    MSM17510
C                                                                       MSM17520
      DIMENSION DP(18,18),CONST(18,18),SP(18),                          MSM17530
     X   CP(18),FN(18),FM(18)                                           MSM17540
      DIMENSION XCO(18,18),MLBL(5)                                      MSM17550
C                                                                       MSM17560
      DATA CMIN,CMAX/50.,90./                                           MSM17570
      DATA P,CONST/648*0./                                              MSM17580
   80 CONTINUE                                                          MSM17590
C                                                                       MSM17600
C                                                                       MSM17610
      NMAX = NNMAX                                                      MSM17620
C                                                                       MSM17630
      PI = 3.1415927                                                    MSM17640
      VALUE=-1.E-9                                                      MSM17650
      IF(TLAT.GT.CMIN)GOTO70                                            MSM17660
      PRINT *,'LATITUDE ',TLAT,' OUT OF MODEL RANGE'                    MSM17670
      RETURN                                                            MSM17680
C                                                                       MSM17690
   70 CONTINUE                                                          MSM17700
C                                                                       MSM17710
C   SEGMENT FOR SCALING GRID-PT. LOCATION TO FIT H-M ELLIPSE.           MSM17720
      TCOL = 90.-TLAT                                                   MSM17730
      TLONG = TLON - PI                                                 MSM17740
      IF(ICNTRL.GE.0) THEN                                              MSM17750
         XX = DXHM(1,IPATT)+AHM(1,IPATT)*(TCOL*COS(TLONG)-DX(1))/A(1)   MSM17760
         YY = DYHM(1,IPATT)+BHM(1,IPATT)*(TCOL*SIN(TLONG)-DY(1))/B(1)   MSM17770
      ELSE                                                              MSM17780
         XX = TCOL*COS(TLONG)                                           MSM17790
         YY = TCOL*SIN(TLONG)                                           MSM17800
      END IF                                                            MSM17810
      XCOL = SQRT(XX**2 + YY**2)                                        MSM17820
      XLAT = 90.-XCOL                                                   MSM17830
      XL = ATAN2(YY,XX)                                                 MSM17840
      XLON = XL+PI                                                      MSM17850
      ALPHA=2./(CMAX-CMIN)                                              MSM17860
      BETA=1.-ALPHA*CMAX                                                MSM17870
      CT=XLAT*ALPHA+BETA                                                MSM17880
      ST=SQRT(1.-CT*CT)                                                 MSM17890
      SPH=SIN(XLON)                                                     MSM17900
      CPH=COS(XLON)                                                     MSM17910
C                                                                       MSM17920
```

```
      IF (P(1,1).NE.0.) GOTO 3
      P(1,1)=1.0
      DP(1,1)=0.0
      SP(1)=0.0
      CP(1)=1.0
      DO 2 N=2,18
      FN(N)=N
      DO 2 M=1,N
      FM(M)=M-1
    2 CONST(N,M)=FLOAT((N-2)**2-(M-1)**2)/
     X    FLOAT((2*N-3)*(2*N-5))
C
    3 SP(2)=SPH
      P(1,1)=1.
      CP(2)=CPH
      DO 4 M=3,NMAX
      SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
    4 CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
C
      VALUE=COEF(1,1)
C
      DO 8 N=2,NMAX
      DO 8 M=1,N
C
      IF (N.NE.M) GOTO 6
      P(N,N)=ST*P(N-1,N-1)
C     DP(N,N)=ST*DP(N-1,N-1)+CT*P(N-1,N-1)
      GOTO 8
    6 IF (N.NE.2) P(N,M)=CT*P(N-1,M)-CONST(N,M)*P(N-2,M)
      IF (N.EQ.2) P(N,M)=CT*P(N-1,M)
C     DP(N,M)=CT*DP(N-1,M)-ST*P(N-1,M)-
C    X    CONST(N,M)*DP(N-2,M)
C
C
    8 CONTINUE
C
      P(1,1)=P(1,1)*XCO(1,1)
C
      VALUE=COEF(1,1)*P(1,1)
      DO 10 N=2,NMAX
      DO 10 M=1,N
      IF (M.EQ.1) GOTO 11
      POL=P(N,M)*XCO(N,M)
      P(M-1,N)=CP(M)*POL
      P(N,M)=SP(M)*POL
      VALUE=VALUE+P(M-1,N)*COEF(M-1,N)+
     X    P(N,M)*COEF(N,M)
      GOTO 10
   11 P(N,M)=P(N,M)*XCO(N,M)
      VALUE=VALUE+P(N,M)*COEF(N,M)
   10 CONTINUE
      IF(ICNTRL.NE.-1) THEN
         VALUE = (PCP/(VMAX(IPATT)-VMIN(IPATT)))*VALUE*1000.
      ELSE
         VALUE = 1000.*VALUE
      END IF
C
C
      RETURN
      END
C
C
      SUBROUTINE LOW(COL,ALO,VLOW,DEQDT,AEQEDG,DXEQEG,VBAR,G)
C
```

```
C  PURPOSE:  RETURN THE ELECTRIC POTENTIAL AT A SPECIFIC LOCATION IN
C     THE LOW-LATITUDE REGION OF THE ELECTRIC FIELD MODEL.
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C  INPUTS:
C    COL            COLATITUDE (RADIANS)
C    ALO            MLT (RADIANS)
C    DEQDT          TIME RATE OF CHANGE OF EQUATORWARD EDGE OF AURORAL
C                   ZONE (DEGREES/HOUR)
C    AEQEDG         RADIUS OF ELLIPSE 3 IN X (SUNWARD) DIRECTION
C    DXEQEG         OFFSET OF ELLIPSE 3 IN X DIRECTION
C    VBAR           AVERAGE POTENTIAL AT LOW LATITUDES
C    G              WEIGHTING FACTOR
C
C
C  OUTPUT:
C    VLOW           ELECTRIC POTENTIAL (VOLTS)
C
C
C
      PI = 3.1415927
      AK = 1.38
      SA = SIN((AEQEDG-DXEQEG)*PI/180.)
      CF = 2.75*1000.*1.576*G*DEQDT*SA/SIN(.4363)
      FAC = .6103*SIN(ALO)-.0154*SIN(2.*ALO)-.021*SIN(3.*ALO)
      FAC = FAC - .1092*COS(ALO)+.1676*COS(2.*ALO)-.0314*COS(3.*ALO)
      VLOW = CF*FAC*(SA/SIN(COL))**AK + VBAR
      RETURN
      END
C
C
C
      SUBROUTINE AURORA(COL,ALO,VAZ,VBAZ,TB,TC,J,JMAX,JWRAP)
C
C  PURPOSE:  PROVIDE THE ELECTRIC POTENTIAL AT A SPECIFIC LOCATION IN
C            THE AURORAL REGION OF THE ELECTRIC FIELD MODEL.
C
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C  INPUTS:
C    COL            COLATITUDE OF POINT
C    ALO            MLT OF POINT
C    VBAZ           AURORAL CONTRIBUTION TO POTENTIAL
C    TB             LOCATION OF BOUNDARY B FOR SPECIFIC LOCAL TIME
C    TC             LOCATION OF BOUNDARY C FOR SPECIFIC LOCAL TIME
C    J              LOCAL TIME INDEX
C    JMAX           MAXIMUM NUMBER OF PTS IN J (LONGITUDE) DIRECTION
C    JWRAP          NUMBER OF PTS OF OVERLAP IN J DIRECTION
C
C
C  OUTPUT:
C    VAZ            ELECTRIC POTENTIAL (VOLTS)
C
C
      PARAMETER (JDIM=51)
      DIMENSION VBAZ(JDIM)
C
C
C
      IF(JDIM.NE.JMAX) THEN
```

```
      WRITE(6,147)JDIM,JMAX
 147  FORMAT(1H ,'JDIM = ',I4,'  LTDIM = ',I4,'IN AURORA.')
      END IF
      DPWIDE = 1.
      DPAMP = 2.0
      PI = 3.1415927
      ALMIN = PI - .5*DPWIDE
      ALMAX = PI + .5*DPWIDE
C
      IF(ALO.GT.ALMIN.AND.ALO.LT.ALMAX) THEN
         PHIP = COS((PI*(ALO-PI))/DPWIDE)
      ELSE
         PHIP = 0.
      END IF
      IF (J.EQ.1) THEN
         JM1 = JMAX - JWRAP
         ELSE
         JM1 = J-1
      END IF
C
      IF (J.EQ.JMAX) THEN
         JP1 = JWRAP + 1
         ELSE
         JP1 = J+1
      END IF
C
C
      DVBDPH = (VBAZ(JP1) - VBAZ(JM1))*PI/FLOAT(JMAX-3)
      DENOM = TC - COL
      ANUM = TC - TB
      IF (DENOM.LT.0.001) DENOM = 0.001
C
      CORR = -DVBDPH*PHIP*DPAMP*6.75*(TC-COL)**2*(COL-TB)/(TC-TB)**3
C
      R = 1.
      DTHETA = (1.25-.75*COS(ALO-.75*PI))*(TC-TB)
      FAC2 = (1.+(TC-COL)**2/DTHETA**2)**(-R)
      FAC3 = (1.+(TC-TB)**2/DTHETA**2)**(-R)
      VAZ = VBAZ(J)*(1.-FAC2)/(1.-FAC3) + CORR
C
      RETURN
      END
C
C
C
      SUBROUTINE REG1(COL,AL,V1,TA,TB,VAA,VBB,DVAA,DVBB)
C
C  PURPOSE:  PROVIDE ELECTRIC POTENTIAL IN REGION 1 OF THE ELECTRIC
C            FIELD MODEL
C
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C
C  INPUTS:
C    COL             COLATITUDE (RADIANS)
C    AL              MLT
C    TA              LOCATION OF BOUNDARY A
C    TB              LOCATION OF BOUNDARY B
C    VAA             ELECTRIC POTENTIAL FOR PT ON BOUNDARY A
C    VBB             ELECTRIC POTENTIAL FOR PT ON BOUNDARY B
C    DVAA            DERIVATIVE OF V FOR PT ON BOUNDARY A
C    DVBB            DERIVATIVE OF V FOR PT ON BOUNDARY B
```

```
C
C
C  OUTPUT:
C    V1             ELECTRIC POTENTIAL FOR A POINT IN REGION 1 (VOLTS)
C
C
      DT = TB - TA
      V1 = VAA + DVAA*(COL-TA)
      V1 = V1 + (3.*(VBB-VAA)-(2.*DVAA+DVBB)*DT)*(COL-TA)**2/DT**2
      V1 = V1 + (2.*(VAA-VBB)+(DVAA+DVBB)*DT)*(COL-TA)**3/DT**3
      RETURN
      END
C
C
      FUNCTION THET(AA,BB,XC,YC,PHI)
C     THIS FUNCTION GIVES SOLUTION TO EQUATION FOR ELLIPSE IN
C     IN FLAT POLAR COORDINATES.  IT SPECIFIES COLATITUDE AS A
C     FUNCTION OF LOCAL TIME.
C
C  INPUTS:
C    AA             RADIUS OF ELLIPSE IN X (SUNWARD) DIRECTION
C    BB             RADIUS OF ELLIPSE IN Y (DUSKWARD) DIRECTION
C    XC             X DISPLACEMENT OF ELLIPSE
C    YC             Y DISPLACEMENT OF ELLIPSE
C    PHI            MLT HOUR ANGLE
C
C
C  OUTPUT:
C    THET           COLATITUDE OF INPUT ELLIPSE AT GIVEN PHI
C
         CPHI = COS(PHI)
         SPHI = SIN(PHI)
         CA = (CPHI/AA)**2 + (SPHI/BB)**2
         CB = -XC*CPHI/AA**2 - YC*SPHI/BB**2
         CC = (XC/AA)**2 + (YC/BB)**2 - 1.
         THET = (-CB+SQRT(CB**2-CA*CC))/CA
      RETURN
      END
C
C
      SUBROUTINE INPUT(NTAPE,COEF,IPATT,XCO,AHM,BHM,DXHM,
     1  DYHM,VMIN, VMAX)
C
C  PURPOSE:  READ IN NECESSARY INPUT FOR THE COMPUTATION OF THE
C            HEPPNER-MAYNARD ELECTRIC FIELD MODEL
C
      DIMENSION COEF(18,18)
      DIMENSION DXHM(2,7),DYHM(2,7)
      DIMENSION XCO(18,18), AHM(2,7),BHM(2,7)
      DIMENSION VMIN(7),VMAX(7),YCO(18,18)
      DIMENSION AAHM(2,7),BBHM(2,7),DDXHM(2,7)
      DIMENSION DDYHM(2,7), VVMAX(7), VVMIN(7)
C
      CHARACTER*50 MLBL
      DATA (YCO( 1,J),J=1,18)/
     X .282095E+00, .488603E+00, .109255E+01, .228523E+01,
     X .468333E+01, .951188E+01, .192265E+02, .387523E+02,
     X .779645E+02, .156658E+03, .314501E+03, .630964E+03,
     X .126523E+04, .253611E+04, .508196E+04, .101809E+05,
     X   .203918E+05,    .408366E+05/
      DATA (YCO( 2,J),J=1,18)/
     X .488603E+00, .488603E+00, .546274E+00, .144531E+01,
     X .331161E+01, .719031E+01, .151999E+02, .316411E+02,
     X .652298E+02, .133599E+03, .272366E+03, .553392E+03,
```

```
     X .112151E+04, .226837E+04, .458082E+04, .923904E+04,             MSM20490
     X   .186151E+05,   .374743E+05/                                    MSM20500
      DATA (YCO( 3,J),J=1,18)/                                          MSM20510
     X .946175E+00, .109255E+01, .546274E+00, .590044E+00,             MSM20520
     X .177013E+01, .440314E+01, .101333E+02, .223736E+02,             MSM20530
     X .481754E+02, .102038E+03, .213661E+03, .443701E+03,             MSM20540
     X .915709E+03, .188083E+04, .384866E+04, .785168E+04,             MSM20550
     X   .159791E+05,   .324537E+05/                                    MSM20560
      DATA (YCO( 4,J),J=1,18)/                                          MSM20570
     X .186588E+01, .228523E+01, .144531E+01, .590044E+00,             MSM20580
     X .625836E+00, .207566E+01, .555021E+01, .134918E+02,             MSM20590
     X .310971E+02, .693209E+02, .151081E+03, .324033E+03,             MSM20600
     X .686782E+03, .144253E+04, .300864E+04, .623988E+04,             MSM20610
     X   .128827E+05,   .264983E+05/                                    MSM20620
      DATA (YCO( 5,J),J=1,18)/                                          MSM20630
     X .370249E+01, .468333E+01, .331161E+01, .177013E+01,             MSM20640
     X .625836E+00, .656382E+00, .236662E+01, .674590E+01,             MSM20650
     X .172496E+02, .414272E+02, .955522E+02, .214328E+03,             MSM20660
     X .471128E+03, .102002E+04, .218269E+04, .462762E+04,             MSM20670
     X   .973844E+04,   .203694E+05/                                    MSM20680
      DATA (YCO( 6,J),J=1,18)/                                          MSM20690
     X .736787E+01, .951188E+01, .719031E+01, .440314E+01,             MSM20700
     X .207566E+01, .656382E+00, .683184E+00, .264596E+01,             MSM20710
     X .798499E+01, .213929E+02, .534153E+02, .127330E+03,             MSM20720
     X .293800E+03, .661878E+03, .146420E+04, .319336E+04,             MSM20730
     X   .688612E+04,   .147131E+05/                                    MSM20740
      DATA (YCO( 7,J),J=1,18)/                                          MSM20750
     X .146845E+02, .192265E+02, .151999E+02, .101333E+02,             MSM20760
     X .555021E+01, .236662E+01, .683184E+00, .707163E+00,             MSM20770
     X .291571E+01, .926339E+01, .259102E+02, .671087E+02,             MSM20780
     X .165101E+03, .391572E+03, .903721E+03, .204248E+04,             MSM20790
     X   .454057E+04,   .996084E+04/                                    MSM20800
      DATA (YCO( 8,J),J=1,18)/                                          MSM20810
     X .292940E+02, .387523E+02, .316411E+02, .223736E+02,             MSM20820
     X .134918E+02, .674590E+01, .264596E+01, .707163E+00,             MSM20830
     X .728927E+00, .317732E+01, .105778E+02, .307916E+02,             MSM20840
     X .825507E+02, .209304E+03, .509767E+03, .120459E+04,             MSM20850
     X   .278052E+04,   .629979E+04/                                    MSM20860
      DATA (YCO( 9,J),J=1,18)/                                          MSM20870
     X .584734E+02, .779645E+02, .652298E+02, .481754E+02,             MSM20880
     X .300971E+02, .172496E+02, .798499E+01, .291571E+01,             MSM20890
     X .728927E+00, .748901E+00, .343190E+01, .119255E+02,             MSM20900
     X .360281E+02, .997819E+02, .260366E+03, .650553E+03,             MSM20910
     X   .157290E+04,   .370647E+04/                                    MSM20920
      DATA (YCO(10,J),J=1,18)/                                          MSM20930
     X .116766E+03, .156658E+03, .133599E+03, .102038E+03,             MSM20940
     X .693209E+02, .414272E+02, .213929E+02, .926339E+01,             MSM20950
     X .317732E+01, .748901E+00, .767395E+00, .368030E+01,             MSM20960
     X .133043E+02, .416119E+02, .118840E+03, .318704E+03,             MSM20970
     X   .816138E+03,   .201755E+04/                                    MSM20980
      DATA (YCO(11,J),J=1,18)/                                          MSM20990
     X .233240E+03, .314501E+03, .272366E+03, .213661E+03,             MSM21000
     X .151081E+03, .955522E+02, .534153E+02, .259102E+02,             MSM21010
     X .105778E+02, .343190E+01, .767395E+00, .784642E+00,             MSM21020
     X .392321E+01, .147120E+02, .475361E+02, .139761E+03,             MSM21030
     X   .384731E+03,   .100877E+04/                                    MSM21040
      DATA (YCO(12,J),J=1,18)/                                          MSM21050
     X .465998E+03, .630964E+03, .553392E+03, .443701E+03,             MSM21060
     X .324033E+03, .214328E+03, .127330E+03, .671087E+02,             MSM21070
     X .307916E+02, .119255E+02, .368030E+01, .784642E+00,             MSM21080
     X .800822E+00, .416119E+01, .161472E+02, .537941E+02,             MSM21090
     X   .162579E+03,   .458849E+03/                                    MSM21100
      DATA (YCO(13,J),J=1,18)/                                          MSM21110
     X .931187E+03, .126523E+04, .112151E+04, .915709E+03,             MSM21120
```

```
     X .686782E+03, .471128E+03, .293800E+03, .165101E+03,
     X .825507E+02, .360281E+02, .133043E+02, .392321E+01,
     X .800822E+00, .816077E+00, .439471E+01, .176082E+02,
     X   .603802E+02,   .187325E+03/
      DATA (YCO(14,J),J=1,18)/
     X .186100E+04, .253611E+04, .226837E+04, .188083E+04,
     X .144253E+04, .102002E+04, .661878E+03, .391572E+03,
     X .209304E+03, .997819E+02, .416119E+02, .147120E+02,
     X .416119E+01, .816077E+00, .830522E+00, .462415E+01,
     X   .190939E+02,   .672889E+02/
      DATA (YCO(15,J),J=1,18)/
     X .371962E+04, .508196E+04, .458082E+04, .384866E+04,
     X .300864E+04, .218269E+04, .146420E+04, .903721E+03,
     X .509767E+03, .260366E+03, .118840E+03, .475361E+02,
     X .161472E+02, .439471E+01, .830522E+00, .844251E+00,
     X   .484985E+01,   .206029E+02/
      DATA (YCO(16,J),J=1,18)/
     X .743510E+04, .101809E+05, .923904E+04, .785168E+04,
     X .623988E+04, .462762E+04, .319336E+04, .204248E+04,
     X .120459E+04, .650553E+03, .318704E+03, .139761E+03,
     X .537941E+02, .176082E+02, .462415E+01, .844251E+00,
     X   .857341E+00,   .507210E+01/
      DATA (YCO(17,J),J=1,18)/
     X .148629E+05, .203918E+05, .186151E+05, .159791E+05,
     X .128827E+05, .973844E+04, .688612E+04, .454057E+04,
     X .278052E+04, .157290E+04, .816138E+03, .384731E+03,
     X .162579E+03, .603802E+02, .190939E+02, .484985E+01,
     X   .857341E+00,   .869857E+00/
      DATA (YCO(18,J),J=1,18)/
     X .297130E+05, .408366E+05, .374743E+05, .324537E+05,
     X .264983E+05, .203694E+05, .147131E+05, .996084E+04,
     X .629979E+04, .370647E+04, .201755E+04, .100877E+04,
     X .458849E+03, .187325E+03, .672889E+02, .206029E+02,
     X   .507210E+01,   .869857E+00/
C
      DATA(AAHM(1,IP),IP=1,7)/17.45,12.13,16.06,13.72,14.79,12.82,15./
      DATA(AAHM(2,IP),IP=1,7)/20.,16.17,18.88,19.31,18.3,16.6,17.93/
      DATA(BBHM(1,IP),IP=1,7)/14.26,13.78,14.31,13.78,14.73,11.70,12.07/
      DATA(BBHM(2,IP),IP=1,7)/16.97,16.7,17.07,17.23,17.93,17.5,16.6/
      DATA(DDXHM(1,IP),IP=1,7)/-2.66,-5.53,-3.19,-.11,-2.45,-1.65,-2.23/
      DATA(DDXHM(2,IP),IP=1,7)/-3.3,-7.45,-4.1,-1.76,-2.77,-2.98,-3.14/
      DATA(DDYHM(1,IP),IP=1,7)/1.6,.8,1.12,.05,.48,3.4,1.22/
      DATA(DDYHM(2,IP),IP=1,7)/1.33,.32,1.54,.85,.69,1.86,.21/
      DATA(VVMAX(IP),IP=1,7)/34007.,55354.,14390.,11287.,9329.,13249.,
     1 13221./
      DATA(VVMIN(IP),IP=1,7)/-42280.,-16003.,-60935.,-16250.,-12947.,
     1 -14428.,-13460./
C
C
      DO 4 IP = 1,7
         DO 3 N=1,2
            AHM(N,IP) = AAHM(N,IP)
            BHM(N,IP) = BBHM(N,IP)
            DXHM(N,IP) = DDXHM(N,IP)
            DYHM(N,IP) = DDYHM(N,IP)
  3      CONTINUE
         VMAX(IP) = VVMAX(IP)
         VMIN(IP) = VVMIN(IP)
  4   CONTINUE
C
C   INITIALIZE COEF FROM UNIT NTAPE
C
C
      DO 1 I=1,18
```

```
         DO 2 J=1,18
           COEF(I,J) = 0.
           XCO(I,J) = YCO(I,J)
 2       CONTINUE
 1     CONTINUE
C
C  OPEN FILE TO GET ELECTRIC FIELD COEFFICIENTS
C
       OPEN(UNIT=NTAPE,FILE='EFCOEF',STATUS='OLD')
C
       DO 81 IA=1,IPATT
       READ(NTAPE,400,ERR=1001,END=1001)MLBL
 1001  CONTINUE
 400   FORMAT(A)
   82  READ(NTAPE,*)NNN,MMM,I,J,CCF
C      WRITE(8,100) IA,NNN,MMM,I,J,CCF
C100   FORMAT(1H ,'IA,NNN,MMM,I,J,CCF = ',5I4,E15.5)
       IF(NNN.EQ.-1) GO TO 81
       IF(I.GT.18.OR.J.GT.18) THEN
         WRITE(6,737) IA,NNN,MMM,I,J
 737     FORMAT(1H ,'OUT OF BOUNDS IN INPUT.IA,NNN,MMM,I,J = ',5I5)
       END IF
       COEF(I,J)=CCF
       GO TO 82
   81  CONTINUE
C      WRITE(8,401)MLBL
C401   FORMAT(1H ,'MLBL = ', 5A10)
       NMAX=NNMAX
C
       CLOSE(NTAPE)
C
       RETURN
C
       END
C
C
       SUBROUTINE BTRACE(LATDIM,LTDIM,ITMDIM,L,FSTOFF,FEQEDG,FDST,
     2              FCLPSE,FTILT,VM,BMIN,XMIN,YMIN,ZMIN,R,P,ALOCT,TETA)
C
C  VERSION 2.02                          DATE: 09.16.89
C
C  PURPOSE:  SUBROUTINE TO CALCULATE B FIELD MODELS APPROPRIATE FOR
C         GEOPHYSICAL CONDITIONS AS SPECIFIED IN BFPAR.  BTRACE
C         CALCULATES VM, BMIN, XMIN, YMIN, AND ZMIN ON THE MSM SPATIAL
C         GRID BY INTERPOLATING BETWEEN PRECOMPUTED MAGNETIC FIELD
C         MODELS AS PARAMETERIZED BY THE VARIABLES OF BFPAR.
C
C  INPUT:
C        LATDIM    NUMBER OF LATITUDINAL GRID POINTS
C        LTDIM     NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)
C        ITMDIM    MAX NUMBER OF TEMPORAL GRID POINTS
C        L         CURRENT TEMPORAL GRID INDEX
C        .....     GEOPHYSICAL PARAMETERS THAT DETERMINE WHICH BFIELD
C                  MODELS TO USE
C        FSTOFF    STANDOFF DIST = AUGPAR(STAND)
C        FEQEDG    EQUATORWARD EDGE OF AURORA = AUGPAR(EQEDGE)
C        FDST      DST = AUGPAR(DST)
C        FCLPSE    COLLAPSE PARAMETER = AUGPAR(CLAPSE)
C        FTILT     TILT PARAMETER = AUGPAR(TILTW)
C        ALOCT     LOCAL TIME HOUR ANGLE (MEASURED EAST FROM NOON)
C        TETA      COLATITUDE OF I GRID LINES (RADIANS)
C
C  OUTPUT:
C        VM        (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
```

```
C        BMIN       B VALUE AT MINIMUM VALUE ALONG FIELD LINE (NT)
C        XMIN       X (GSM) VALUE AT WHICH BMIN OCCURS
C        YMIN       Y (GSM) VALUE AT WHICH BMIN OCCURS
C        ZMIN       Z (GSM) VALUE AT WHICH BMIN OCCURS
C        R          R SQRT(X**2+Y**2+Z**2) AT WHICH BMIN OCCURS
C        P          HOUR ANGLE (RADIANS) MEASURED EASTWARD FROM NOON
C
C    WORK:
C        WORK       WORK ARRAY (NEEDED FOR BFIELD RETRIEVAL AND
C                     INTERPOLATION)
C
C    REFERENCE:
C        FIVE DIMENSIONAL INTERPOLATION IS BASED ON MULTILINEAR
C          INTERPOLATION SCHEME GIVEN BY PRESS ET AL. IN
C          NUMERICAL RECIPES, CAMBRIDGE UNIVERSITY PRESS, 1987,PP 95-97
C
C
C    PROGRAMMER:  R.W. SPIRO
C
C
      PARAMETER(IIDIM=62,JJDIM=51)
C
      COMMON /GRID/DLAM,DPSI,RI,JWRAP
C
      LOGICAL*1 BFEXST(2,2,2,2,2)
C
      DIMENSION WORK(IIDIM,JJDIM,2,2,2,2,2)
      DIMENSION VM(LATDIM,LTDIM,ITMDIM)
      DIMENSION BMIN(LATDIM,LTDIM),XMIN(LATDIM,LTDIM)
      DIMENSION YMIN(LATDIM,LTDIM),ZMIN(LATDIM,LTDIM)
      DIMENSION ALOCT(LATDIM,LTDIM),R(LATDIM,LTDIM,ITMDIM)
      DIMENSION P(LATDIM,LTDIM,ITMDIM)
      DIMENSION BFLIM(2,5),BFPAR(5)
      DIMENSION T(2,5),N(5)
      DIMENSION SUMC(IIDIM,JJDIM)
      DIMENSION TETA(LATDIM)
C
      DATA BFBUSH /1.E20/
C
C
C    CHECK DIMENSIONS
      IF(IIDIM.NE.LATDIM.OR.JJDIM.NE.LTDIM) THEN
         WRITE(6,*) 'DIMENSION ERROR IN BTRACE'
         WRITE(6,*) 'IIDIM= ',IIDIM,' SHOULD EQUAL LATDIM= ',LATDIM
         WRITE(6,*) 'JJDIM= ',JJDIM,' SHOULD EQUAL LTDIM= ',LTDIM
         WRITE(6,*) 'STOPPING PGM IN BTRACE; CHECK DIMENSIONS'
         STOP
      END IF
C
C
C
      PI=ATAN2(0.,-1.)
C
C    SET UP BFPAR VECTOR FOR CALLING GETMAT
      BFPAR(1)=FSTOFF
      BFPAR(2)=FTILT
      BFPAR(3)=FEQEDG
      BFPAR(4)=FDST
      BFPAR(5)=FCLPSE
C
C
C
C    INITIALIZE BFIELD ARRAYS
C
```

```
      DO 15 J=1,LTDIM
         DO 16 I=1,LATDIM
            VM(I,J,L)=0.
            BMIN(I,J)=0.
            XMIN(I,J)=0.
            YMIN(I,J)=0.
            ZMIN(I,J)=0.
            SUMC(I,J)=0.
   16    CONTINUE
   15 CONTINUE
C
      WRITE (6,*) 'GETMAT IS REQUESTING THE FOLLOWING B-MATRICES:'
C
C  LOOP OVER FIVE FLAVORS OF B FIELD ARRAYS
      DO 20 NN=1,5
         IWANT=NN
C
C  CALL GETMAT TO READ APPROPRIATE ARRAYS FOR INTERPOLATION
C
         CALL GETMAT(IWANT,LATDIM,LTDIM,BFPAR,WORK,BFLIM,BFEXST)
         CALL RMVBSH(WORK,LATDIM,LTDIM)
C
C  SET UP WEIGHTING PARAMETERS FOR INTERPOLATION
C
         DO 10 II=1,5
            T(2,II)=(BFPAR(II)-BFLIM(1,II))/(BFLIM(2,II)-BFLIM(1,II))
            T(1,II)=1.-T(2,II)
   10    CONTINUE
C
         IF(IWANT.EQ.1) THEN
            WRITE(6,*) 'T ARRAY',T
         END IF
C
C
C  MAIN INTERPOLATION LOOPS
C
      DO 21 N1=1,2
         N(1)=N1
         DO 22 N2=1,2
            N(2)=N2
            DO 23 N3=1,2
               N(3)=N3
               DO 24 N4=1,2
                  N(4)=N4
                  DO 25 N5=1,2
                     N(5)=N5
C
                     COEFF=1.
C
                     DO 30 M=1,5
                        COEFF=COEFF*T(N(M),M)
   30                CONTINUE
C
C  CHECK TO SEE IF B FIELD MODEL EXISTS
C
C                     WRITE(6,*) 'N1,N2,N3,N4,N5',N1,N2,N3,N4,N5
C                     WRITE(6,*) 'COEFF',COEFF
C                     WRITE(6,*) 'BFEXST',BFEXST(N1,N2,N3,N4,N5)
                     IF(.NOT.BFEXST(N1,N2,N3,N4,N5)) THEN
                        COEFF=0.
                     END IF
C
C
C
```

```
C  LOOP OVER SPATIAL GRID
C
                  IF(IWANT.EQ.1) THEN
                     DO 40 J=1,LTDIM
                        DO 50 I=1,LATDIM
                         IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH) THEN
                           XMIN(I,J)=XMIN(I,J)+COEFF*
     2                                 WORK(I,J,N1,N2,N3,N4,N5)
                         ENDIF
   50                   CONTINUE
   40                CONTINUE
C                     WRITE(6,*)' XMIN VALUES'
C                     WRITE(6,*)(XMIN(I,33),I=20,26)
C                     WRITE(6,*)(XMIN(I,12),I=20,26)
C                     WRITE(6,*)' WORK VALUES '
C                     WRITE(6,*)(WORK(I,33,N1,N2,N3,N4,N5),I=20,26)
C                     WRITE(6,*)(WORK(I,12,N1,N2,N3,N4,N5),I=20,26)
C                     WRITE(6,*)N1,N2,N3,N4,N5
C
                  ELSE IF(IWANT.EQ.2) THEN
                     DO 60 J=1,LTDIM
                        DO 70 I=1,LATDIM
                         IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH) THEN
                           YMIN(I,J)=YMIN(I,J)+COEFF*
     2                                 WORK(I,J,N1,N2,N3,N4,N5)
                         END IF
   70                   CONTINUE
   60                CONTINUE
C
                  ELSE IF(IWANT.EQ.3) THEN
                     DO 80 J=1,LTDIM
                        DO 90 I=1,LATDIM
                         IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH) THEN
                           ZMIN(I,J)=ZMIN(I,J)+COEFF*
     2                                 WORK(I,J,N1,N2,N3,N4,N5)
                         END IF
   90                   CONTINUE
   80                CONTINUE
C
                  ELSE IF(IWANT.EQ.4) THEN
                     DO 100 J=1,LTDIM
                        DO 110 I=1,LATDIM
                         IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH) THEN
                           BMIN(I,J)=BMIN(I,J)+COEFF*
     2                                 WORK(I,J,N1,N2,N3,N4,N5)
                         END IF
  110                   CONTINUE
  100                CONTINUE
C
                  ELSE
                     DO 120 J=1,LTDIM
                        DO 130 I=1,LATDIM
                         IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH) THEN
                           VM(I,J,L)=VM(I,J,L)+COEFF*
     2                                 WORK(I,J,N1,N2,N3,N4,N5)
                           SUMC(I,J)=SUMC(I,J)+COEFF
                         END IF
  130                   CONTINUE
  120                CONTINUE
                  ENDIF
   25             CONTINUE
   24          CONTINUE
   23       CONTINUE
   22    CONTINUE
```

```
   21    CONTINUE
   20 CONTINUE
C
C     CALL OUTP(SUMC,IIDIM,JJDIM,10,IIDIM,1,3,JJDIM-3,1,1.,'SUMC',6,132)
C
C  CORRECT FOR MISSING B FIELD MATRICES
C
      DO 140 J=1,LTDIM
         DO 150 I=1,LATDIM
            IF(SUMC(I,J).GT.1.E-10) THEN
               VM(I,J,L)=VM(I,J,L)/SUMC(I,J)
               BMIN(I,J)=BMIN(I,J)/SUMC(I,J)
               XMIN(I,J)=XMIN(I,J)/SUMC(I,J)
               YMIN(I,J)=YMIN(I,J)/SUMC(I,J)
               ZMIN(I,J)=ZMIN(I,J)/SUMC(I,J)
            END IF
  150    CONTINUE
  140 CONTINUE
C
C
C
C
C  ROTATE BFIELD ARRAYS TO MATCH GRID ORIENTATION
C    NOTE: IN THE CALLS TO BFGYRO, R IS BEING USED AS A WORK ARRAY
C
      CALL BFGYRO(LATDIM,LTDIM,VM(1,1,L),R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,BMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,XMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,YMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,ZMIN,R(1,1,L),ALOCT)
C
C
      DO 160 J=1,LTDIM
         DO 170 I=1,LATDIM
            R(I,J,L)=SQRT(XMIN(I,J)**2+YMIN(I,J)**2+ZMIN(I,J)**2)
C
            IF(XMIN(I,J).EQ.0..AND.YMIN(I,J).EQ.0.) THEN
               P(I,J,L)=0.
            ELSE
               P(I,J,L)=ATAN2(YMIN(I,J),XMIN(I,J))
               IF(P(I,J,L).LT.0.) THEN
                  P(I,J,L)=P(I,J,L)+2.*PI
               END IF
            END IF
C
  170    CONTINUE
  160 CONTINUE
C
      RETURN
      END
      SUBROUTINE BFGYRO(LATDIM,LTDIM,ARRAY,WORK,ALOCT)
C
C  VERSION 1.00                         DATE: 08.29.89
C
C  PURPOSE:  SUBROUTINE TO ROTATE FIXED GRID B FIELD ARRAYS TO MATCH
C           ROTATING COORDINATE SYSTEM
C
C  NOTE:  THIS SUBROUTINE USES ARRAY AS BOTH AN INPUT AND OUTPUT
C         ARRAY.  ON INPUT, ARRAY CONTAINS THE UNROTATED DATA; ON
C         OUTPUT, IT CONTAINS THE APPROPRIATELY ROTATED DATA.
C
C  INPUTS:
C    LATDIM         NUMBER OF LATITUDINAL GRID LINES
C    LTDIM          NUMBER OF LOCAL TIME (LONGITUDE) GRID LINES
```

```
C     ARRAY          UNROTATED MAGNETIC FIELD ARRAY
C     WORK           INTERNAL WORKING ARRAY
C     ALOCT          LOCAL TIME HOUR ANGLE (MEASURED EAST FROM NOON)
C                    (RADIANS)
C
C   OUTPUT:
C     ARRAY          ROTATED MAGNETIC FIELD ARRAY
C
C
      COMMON /GRID/DLAM,DPSI,RI,JWRAP
C
      DIMENSION ARRAY(LATDIM,LTDIM),WORK(LATDIM,LTDIM)
      DIMENSION ALOCT(LATDIM,LTDIM)
C
      PI=ATAN2(0.,-1.)
C
C   CALCULATE ROTATION OF COORDINATE SYSTEM
C
      ROT=ALOCT(1,3)*(LTDIM-JWRAP)/(2.*PI)
      JROT=ROT
      F=ROT-FLOAT(JROT)
C
      DO 10 J=1,LTDIM
         J1=J+JROT
         J1=MOD(J1,LTDIM-JWRAP)
         IF(J1.EQ.0) J1=LTDIM-JWRAP
         J2=J1+1
         DO 20 I=1,LATDIM
            WORK(I,J)=(1.-F)*ARRAY(I,J1)+F*ARRAY(I,J2)
   20    CONTINUE
   10 CONTINUE
C
C
C   PUT ROTATED ARRAY BACK INTO ARRAY
C
      DO 30 J=1,LTDIM
         DO 40 I=1,LATDIM
            ARRAY(I,J)=WORK(I,J)
   40    CONTINUE
   30 CONTINUE
C
C
      RETURN
      END
C
C
      SUBROUTINE PPTM(TSTART,TSTOP,DTNOM,SCTFRC,IFLAV,BNDMAX,
     1          ALPHA,BETA,BIR,ITMMAX,IDIM,JDIM,IEDIM,ITMDIM,IRDIM,
     2          NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
     3          THRMAT,TIMTAG,ALAM,ALMDEL,VM,R,P,VMLOSS,V,ETABEG,
     4          ETABND,BNDLOC,ETA,EFLUX,FLXSUM,EAVG,LL,RPP,FKP,
     5          IEMAX,THRSH,TSHENG,TSHING,IRDK,INRGDK,ISOLDK,IONDK,
     6          DKTIME)
C
C   VERSION 1.00                DATE: 01.07.88
C           1.10                      09.12.89
C           1.20                      10.10.89
C           1.30                      05.09.90
C
C   PURPOSE:  SUBROUTINE TO TRACE PARTICLE TRAJECTORIES FROM EACH GRID
C           POINT FOR EACH PARTICLE SPECIES BACK IN TIME FROM
C           TIME=TSTART TO TIME=TSTOP.  LOSS BY PRECIPITATION INTO
C           THE LOSS CONE IS CALCULATED FOR EACH TEST PARTICLE ALONG
C           ITS TRAJECTORY.  PARTICLE DISTRIBUTIONS AND PRECIPITATING
```

```
C          ENERGY FLUXES ARE CALCULATED AT EACH GRID POINT FOR TIME      MSM25610
C          TSTART FROM INITIAL AND BOUNDARY VALUES AT TIME TSTOP.        MSM25620
C          NOTE: TSTART.GT.TSTOP                                         MSM25630
C                                                                        MSM25640
C  INPUT:                                                                MSM25650
C      TSTART       TIME AT WHICH TO BEGIN PARTICLE TRACEBACK            MSM25660
C      TSTOP        TIME AT WHICH TO END PARTICLE TRACEBACK              MSM25670
C      DTNOM        NOMINAL TIME STEP TO BEGIN TRACES.  DTNOM.LT.0       MSM25680
C                   TO TRACE BACK IN TIME.  PPTM WILL ADJUST             MSM25690
C                   TIME STEP AS NEEDED.                                 MSM25700
C      SCTFRC       SCATTERING FRACTION OF STRONG PITCH ANGLE DIFFUSION MSM25710
C      IFLAV        SPECIES DEFINITION INDEX. IFLAV=1 FOR ELECTRONS      MSM25720
C                   IFLAV=2 FOR H+ IONS; IFLAV=3 FOR O+ IONS             MSM25730
C      BNDMAX       MAX I VALUE OF BNDY IN TIME INTERVAL                 MSM25740
C      ALPHA        LATITUDINAL GRID SPACING VECTOR                      MSM25750
C      BETA         LOCAL TIME GRID SPACING VECTOR                       MSM25760
C      BIR          IONSOPHERIC MAGNETIC FIELD STRENGTH                  MSM25770
C      ITMMAX       NUMBER OF TEMPORAL GRID POINTS USED                  MSM25780
C      IDIM         NUMBER OF LATITUDINAL GRID POINTS                    MSM25790
C      JDIM         NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)  MSM25800
C      IEDIM        NUMBER OF ENERGY SPECIES                             MSM25810
C      ITMDIM       MAX NUMBER OF TEMPORAL GRID POINTS                   MSM25820
C      IRDIM        R DEPTH IN FLXMAT ARRAY                              MSM25830
C      NRGDIM       ENERGY DEPTH IN FLXMAT ARRAY                         MSM25840
C      KPDIM        KP DEPTH IN FLXMAT ARRAY                             MSM25850
C      KPPLUS       AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)           MSM25860
C      ISPDIM       NUMBER OF MASS SPECIES IN FLXMAT ARRAY               MSM25870
C      FLXR         ALOG10(R) VALUES FOR WHICH FLXMAT ARRAY IS           MSM25880
C                   CALCULATED                                           MSM25890
C      FLXNRG       ALOG10(ENERGY) VALUES FOR WHICH FLXMAT ARRAY IS      MSM25900
C                   CALCULATED                                           MSM25910
C      FLXKP        KP VALUES FOR WHICH FLXMAT ARRAY IS CALCULATED       MSM25920
C      FLXMAT       ALOG10(FLUX) FOR A GIVEN R,ENERGY, AND KP            MSM25930
C      THRMAT       ARRAY GIVING LOG OF THRESHOLD FLUXES                 MSM25940
C      TIMTAG       VECTOR GIVING UNIVERSAL TIMES FOR WHICH DATA ARRAYS MSM25950
C                   ARE GIVEN. (SECONDS FROM BEGINNING OF START DAY)     MSM25960
C      ALAM         ENERGY INVARIANT OF PARTICLE                         MSM25970
C      ALMDEL       ENERGY INVARIANT SPACING                             MSM25980
C      VM           (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3         MSM25990
C      R            DISTANCE TO BMIN PT (RE)                             MSM26000
C      P            HOUR ANGLE (EASTWARD FROM NOON) (RADIANS)            MSM26010
C      VMLOSS       AUX. LOSS ARRAY= (SINI/BIR)*VM**2                    MSM26020
C      V            ELECTRIC POTENTIAL ARRAY (VOLTS)                     MSM26030
C      ETABEG       INITIAL ETA DISTRIBUTION (IE., AT T=TSTOP)           MSM26040
C      ETABND       ETA DISTRIBUTION ON BOUNDARY (DEPENDS ON TIME)       MSM26050
C      BNDLOC       OUTER BNDY OF CALCULATION (DEPENDS ON TIME)          MSM26060
C      LL           CURRENT TEMPORAL GRID INDEX                          MSM26070
C      RPP          NOMINAL RADIUS OF PLASMAPAUSE                        MSM26080
C      FKP          CURRENT KP VALUE                                     MSM26090
C      IEMAX        NUMBER OF ENERGY CHANNELS BEING USED                 MSM26100
C      THRSH        VECTOR GIVING FULL TRACE/DEFAULT FLUX INFORMATION    MSM26110
C      TSHENG       THRESHOLD ENERGY FOR SWITCHING FROM FULL TRACE       MSM26120
C                   TO EMPIRICAL MODEL FOR ELECTRONS                     MSM26130
C      TSHING       THRESHOLD ENERGY FOR SWITCHING FROM FULL TRACE TO    MSM26140
C                   EMPIRICAL MODEL FOR IONS                             MSM26150
C      IRDK         R DIMENSION FOR DKTIME ARRAY                         MSM26160
C      INRGDK       ENERGY DIMENSION FOR DKTIME ARRAY                    MSM26170
C      ISOLDK       SUN SPOT NUMBER DIMENSION FOR DKTIME ARRAY           MSM26180
C      IONDK        NUMBER OF ION SPECIES IN DKTIME ARRAY                MSM26190
C      DKTIME       TABLE OF ION CHARGE EXCHANGE DECAY TIMES             MSM26200
C                                                                        MSM26210
C  OUTPUT:                                                               MSM26220
C      ETA          COMPUTED ETA DISTRIBUTION CORRESPONDING TO T=TSTART MSM26230
C      EFLUX        PRECIPITATING ENERGY FLUX FOR EACH SPECIES           MSM26240
```

```
C       FLXSUM        TOTAL PRECIPITATING ENERGY FLUX FOR: (1) ELECTRONS;
C                            (2) H+ IONS; AND (3) O+ IONS
C       EAVG          AVERAGE ENERGY OF PRECIPITATING PARTICLES FOR:
C                            (1) ELECTRONS; (2) H+ IONS; AND (3) O+ IONS
C
C  PROGRAMMER: R. W. SPIRO
C
C
C
C
      PARAMETER(IKDIM=3,IIDIM=62,JJDIM=51,KKDIM=30,MDIM=1000)
      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      COMMON /MOVE/ IBEG,ISTOP,JBEG,JSTOP,IEBEG,IEEND
      COMMON /SUN/ SSN
      COMMON/TMP6/III,JJJ
C
      DIMENSION ALPHA(IDIM), BETA(IDIM), BIR(IDIM,JDIM)
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM),TIMTAG(ITMDIM)
      DIMENSION R(IDIM,JDIM,ITMDIM),P(IDIM,JDIM,ITMDIM)
      DIMENSION IFLAV(IEDIM),BNDLOC(JDIM,ITMDIM),THRSH(IEDIM)
      DIMENSION SCTFRC(IEDIM,JDIM,ITMDIM),RPP(ITMDIM)
      DIMENSION BNDMAX(JDIM)
      DIMENSION VMLOSS(IDIM,JDIM,ITMDIM),ETABEG(IDIM,JDIM,IEDIM)
      DIMENSION ETABND(JDIM,IEDIM,ITMDIM),ETA(IDIM,JDIM,IEDIM)
      DIMENSION EFLUX(IDIM,JDIM,IEDIM)
      DIMENSION FLXSUM(IDIM,JDIM,IKDIM), EAVG(IDIM,JDIM,IKDIM)
      DIMENSION SUM1(IKDIM),SUM2(IKDIM)
      DIMENSION XMASS(IKDIM)
      DIMENSION ALAM(IEDIM),ALMDEL(IEDIM)
C
      DIMENSION FLXRFF(IIDIM,JJDIM,KKDIM)
      DIMENSION DENSUM(IIDIM,JJDIM,IKDIM)
C
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)
      DIMENSION THRMAT(IRDIM,NRGDIM,KPDIM)
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)
C
      DIMENSION TM(MDIM),BIM(MDIM),BJM(MDIM),WEAK(MDIM),STRONG(MDIM)
      DIMENSION RR(MDIM),PP(MDIM),VVM(MDIM)
C
      DATA XMASS /9.11E-31,1.67E-27,2.67E-26/
      DATA EPS,BBISCL,BBJSCL /.01,1.,1./
      DATA VMGEO/7.0/
C
C
C
C  CHECK DIMENSION COMPATIBILITY
      IF(IIDIM.NE.IDIM.OR.JJDIM.NE.JDIM.OR.KKDIM.LT.IEMAX) THEN
         WRITE(6,*) 'DIMENSION COMPATIBILITY PROBLEMS IN PPTM'
         WRITE(6,*) 'IIDIM= ',IIDIM,' SHOULD BE EQUAL TO IDIM= ',IDIM
         WRITE(6,*) 'JJDIM= ',JJDIM,' SHOULD BE EQUAL TO JDIM= ',JDIM
         WRITE(6,*)'IEMAX= ',IEMAX,' SHOULD BE LESS THAN KKDIM= ',KKDIM
         WRITE(6,*) 'STOPPING PGM IN PPTM'
         STOP
      END IF
C
C
      PI=ATAN2(0.,-1.)
C
C
```

```
C  CALCULATE REFERENCE FLUX ARRAY FOR TIME=TSTART
      CALL SETREF(IDIM,JDIM,IEDIM,ITMDIM,ITMMAX,IRDIM,NRGDIM,KPDIM,
     2            KPPLUS,ISPDIM,TIMTAG,TSTART,IFLAV,VM,R,P,BNDLOC,ALAM,
     3            ALMDEL,FLXMAT,FLXR,FLXNRG,FLXKP,FLXREF,IEMAX)
C
C  DEBUG PRINTOUT
C     DO 400 IE=1,IEMAX
C        WRITE(6,*) 'IE= ',IE
C        CALL OUTP(FLXREF(1,1,IE),IDIM,JDIM,20,IDIM,1,3,JDIM-1,1,0.,
C    2             'FLXREF',6,132)
C 400 CONTINUE
C
C
C
C
         JMID=(JDIM-3)/2+3
         ETAEQK = 0.
C
C  INITIALIZE FLXSUM AND DENSUM
      DO 100 I=1,IDIM
      DO 100 J=1,JDIM
      DO 110 IK=1,IKDIM
C  IK=1 FOR ELECTRONS; IK=2 FOR H+ IONS; IK=3 FOR O+ IONS
         FLXSUM(I,J,IK)=0.
         DENSUM(I,J,IK)=0.
  110 CONTINUE
C
C  INITIALIZE EFLUX
      DO 120 IE=1,IEMAX
         EFLUX(I,J,IE)=0.
  120 CONTINUE
C
  100 CONTINUE
C
C
C  DO 10 LOOP GOES OVER PARTICLE SPECIES TO BE TRACED
C
      DO 10 IE=IEBEG,IEEND
C
C  FOR ELECTRONS WITH ENERGY GREATER THAN TSHENG KEV AT
C   GEOSYNCHRONOUS ORBIT, USE EMPIRICAL FLUXES AND DISREGARD FULL
C   TRACEBACKS. SIMILARLY FOR IONS GREATER THAN TSHING KEV
C
        HIENG=(ABS(ALAM(IE))*VMGEO)/1000.0
        IF((HIENG.GE.TSHENG).AND.(IFLAV(IE).EQ.1))THEN
           CALL FLXDFL(LL,FKP,R,P,ALAM,VM,IRDIM,NRGDIM,KPDIM,KPPLUS,
     1      ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,ALMDEL,IDIM,JDIM,
     2      IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,IE,IE)
           THRSH(IE)=TSHENG
           IEUSE=IE
           GO TO 10
        ENDIF
C
        IF((HIENG.GE.TSHING).AND.(IFLAV(IE).NE.1))THEN
           CALL FLXDFL(LL,FKP,R,P,ALAM,VM,IRDIM,NRGDIM,KPDIM,KPPLUS,
     1       ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,ALMDEL,IDIM,JDIM,
     2       IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,IE,IE)
           THRSH(IE)=TSHING
           IEUSE=IE
           GO TO 10
        ENDIF
C
         IK=IFLAV(IE)
         ISP=IK
```

```
C
C      TIMING INFO FOR TOTAL K VALUE COMP. TIME
C
c        CALL DTIME(ITKST)
C
         IEUSE=IE
         AALAM=ALAM(IE)
C
C  DO 20 LOOP GOES OVER GRID POINTS
C
      DO 20 I=IBEG,ISTOP
         II=1
c        CALL DTIME(ITRCST)
      DO 24 J=JBEG,JSTOP
         III=I
         JJJ=J
         JJ=J
         BBT=TNORML(TSTART,TIMTAG,ITMMAX)
         IBT=BBT
         BNDJT=(1.-BBT+REAL(IBT))*BNDLOC(JJ,IBT)
     1          +(BBT-REAL(IBT))*BNDLOC(JJ,IBT+1)
C
         IF(REAL(I).LT.BNDJT) THEN
            ETA(I,J,IE)=(1.-BBT+REAL(IBT))*ETABND(J,IE,IBT)
     1                  +(BBT-REAL(IBT))*ETABND(J,IE,IBT+1)
            GO TO 24
         END IF
C
C
C  TIME LOOP HAS POSSIBILITY FOR VARYING TIME STEPS.  STOP TRACING
C  PARTICLE TRAJECTORY WHEN (1) BOUNDARY IS CROSSED OR (2) TSTOP IS
C  REACHED.
C
C  INITIALIZATION
C
         T=TSTART
         IT=NINT(T)
         DTNEXT=DTNOM
         FXLOSS=0.
         FXL2=0.
         BBI=I
         BBJ=J
         BBT=TNORML(T,TIMTAG,ITMMAX)
         M=1
         TM(M)=T
         BIM(M)=BBI
         BJM(M)=BBJ
         RR(M)=G3NTRP(R,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
         PP(M)=G3TRPA(P,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
         VVM(M)=G3NTRP(VM,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
         RPPM=G3NTRP(RPP,ITMDIM,1,1,BBT,1.,1.)
         ENRGM=ABS(AALAM)*VVM(M)
C
        IF(ISP.EQ.1) THEN
          FZERO=G3NTRP(SCTFRC,IEDIM,JDIM,ITMDIM,FLOAT(IE),BBJ,BBT)
          RATEF=(4.45E-17)*SQRT(ABS(AALAM)/XMASS(IK))*G3NTRP(VMLOSS,
     2          IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
          STRONG(M)=FZERO*RATEF
          WEAK(M)=WKRATE(RR(M),RPPM,PP(M),ENRGM)
        ELSE
          STRONG(M)=0.
          WEAK(M)=CEXRAT(ISP,ENRGM,RR(M),SSN,DKTIME,IRDK,INRGDK,
     2                   ISOLDK,IONDK)
        END IF
```

```
C
C
C  DEBUG 1.5.90
C     WRITE(6,789)
C 789 FORMAT(1X,'****************************************************')
C     WRITE(6,790) M,TM(M),BIM(M),BJM(M),RR(M),PP(M),VVM(M),RPPM,ENRGM
C 790 FORMAT(1X,'M=',I3,2X,'T=',F8.0,2X,'BI=',F6.3,2X,'BJ=',F6.3,2X,
C    2 'R=',F6.3,2X,'P=',F8.3,2X,'VM=',F8.3,2X,'RPP=',F6.3,2X,
C    3 'ENRG=',1PE12.3)
C     WRITE(6,791) M,T,RATEF,STRONG(M),WEAK(M)
C 791 FORMAT(1X,'DEBUG 1.5.90   M=',I5,2X,'T=',F10.0,2X,
C    2        'RATEF=',1PE12.3,2X,'STRONG LOSS RATE=',E12.3,2X,
C    3        'WEAK LOSS RATE=',E12.3)
C
C  BEGINNING OF TIME LOOP
C
   30 CONTINUE
C
C  MOVE PARTICLE A TIME STEP
C
        TOLD=T
        BIOLD=BBI
        BJOLD=BBJ
        DTTRY=DTNEXT
C
        IF(DTNOM.GT.0..AND.DTTRY.GT.(TSTOP-TOLD)) DTTRY=TSTOP-TOLD
        IF(DTNOM.LT.0..AND.DTTRY.LT.(TSTOP-TOLD)) DTTRY=TSTOP-TOLD
C
        IF(BIOLD.GT.REAL(IDIM)-1.) THEN
           ETA(I,J,IE)=0.
           GO TO 24
        END IF
        CALL MOVER(T,DTTRY,BIOLD,BJOLD,EPS,BBISCL,BBJSCL,DTUSED,DTNEXT,
     1      AALAM,V,VM,TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM,BBI,BBJ)
C
C  DEBUG PRINTOUT 12.29.89
C     WRITE(6,888) IE,T,DTTRY,BIOLD,BJOLD,DTUSED,BBI,BBJ
C 888 FORMAT(1X,'IE=',I2,2X,'T=',F10.2,2X,'DTTRY=',F5.0,2X,'BIOLD=',
C    2       F6.2,2X,'BJOLD=',F6.2,2X,'DTUSED=',F5.0,2X,'NEW BI=',F6.2,
C    3       2X,'NEW BJ=',F6.2)
C
C
C
C
        IT=IT+NINT(DTUSED)
        T=REAL(IT)
        BBT=TNORML(T,TIMTAG,ITMMAX)
C
        M=M+1
        BIM(M)=BBI
        BJM(M)=BBJ
        CALL MODFIX(BJM(M),JDIM,JWRAP)
        TM(M)=T
C
        RR(M)=G3NTRP(R,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
        PP(M)=G3TRPA(P,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
        VVM(M)=G3NTRP(VM,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
        RPPM=G3NTRP(RPP,ITMDIM,1,1,BBT,1.,1.)
C
C
C
C
        ENRG=ABS(ALAM(IE))*VVM(M)
        FZERO=G3NTRP(SCTFRC,IEDIM,JDIM,ITMDIM,FLOAT(IE),BBJ,BBT)
```

```
C
C CALCULATE LOSS RATE THIS TIME STEP
C
      IF(ISP.EQ.1) THEN
C
C                 CALCULATE LOSS RATE FOR ELECTRONS
C
C FIRST, CALCULATE STRONG PITCH ANGLE SCATTERING RATE
        RATEF=(4.45E-17)*SQRT(ABS(AALAM)/XMASS(IK))*G3NTRP(VMLOSS,
     2             IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
        STRONG(M)=FZERO*RATEF
C
C NEXT, CALCULATE WEAK LOSS RATE
        WEAK(M)=WKRATE(RR(M),RPPM,PP(M),ENRG)
C
C
      ELSE
C
C        CALCULATE LOSS RATE FOR IONS
C          STRONG IS SET TO ZERO FOR IONS
C          WEAK IS USED AS AN ALIAS FOR CHARGE EXCHANGE LOSS RATE
C          SSN IS SUNSPOT NUMBER
C          ISP IS SPECIES NUMBER  (2 FOR H+, 3 FOR O+)
C
        WEAK(M)=CEXRAT(ISP,ENRG,RR(M),SSN,DKTIME,IRDK,INRGDK,
     2                  ISOLDK,IONDK)
C
C
C
        STRONG(M)=0.
C
      END IF
C
C
C
C
C CORRECT J MODULUS OF BBJ
C
      CALL MODFIX(BBJ,JDIM,JWRAP)
C
C CHECK IF PARTICLE HAS PASSED EITHER TIME OR SPATIAL BOUNDARY
C   IF SO, EXIT TIME LOOP
C
C
C
      IEXIT=0
C
C IF TIME BOUNDARY IS CROSSED, SET IEXIT=1 AND INTERPOLATE
C   TO FIND ETABEG AT LOCATION (BBI,BBJ)
C
  25 CONTINUE
      CALL TCHK(T,TSTOP,DTNOM,BBI,BBJ,IEUSE,ETABEG,IDIM,JDIM,
     1             IEDIM,ETANIL,IEXIT)
C
      IF(IEXIT.EQ.0) THEN
C                 CHECK TO SEE IF SPATIAL BNDY IS CROSSED
C
        CALL BNDCHK(T,IEUSE,BBI,BBJ,BIOLD,BJOLD,TIMTAG,ETABND,
     1         BNDMAX,BNDLOC,JDIM,IEDIM,ITMDIM,ITMMAX,JWRAP,ETANIL,IEXIT)
C
      END IF
C
C
      IF(IEXIT.EQ.0) THEN
```

```
            GO TO 30
C                     FOR NEXT TIME STEP
         ELSE
C             CALCULATE ETA(I,J,IE)
C
C
               DO 300 N=M,2,-1
C
C  CALCULATE EQUILIBRIUM ETA AT PT (N-1)
                  BT=TNORML(TM(N-1),TIMTAG,ITMMAX)
                  BIT=BIM(N-1)
                  BJT=BJM(N-1)
                  BIE=FLOAT(IE)
                  ENRG=ABS(ALAM(IE))*VVM(N-1)
                  FKPEQ=-(KPDIM+1)
                  FLXEQK=FLXVAL(ISP,FKPEQ,RR(N-1),PP(N-1),ENRG,IRDIM,
     2                   NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
     3                   FLXNRG,FLXKP)
C
                  IF(ISP.EQ.1) THEN
C
                     ETAEQK=FLXEQK*ALMDEL(IE)/7.392E-16/
     2                   SQRT(ABS(ALAM(IE)))/VVM(N-1)
C
                  ELSE
                     IF(ISP.EQ.2) AWT=1.
                     IF(ISP.EQ.3) AWT=16.
                     ETAEQK=(FLXEQK*ALMDEL(IE)*SQRT(AWT))/(1.731E-17*
     2                       SQRT(ABS(ALAM(IE))))/VVM(N-1)
                  END IF
C
C
C
                IF(ISP.EQ.1) THEN
C                                  CALCULATE THRESHOLD ETA
C                    SET UP INTERPOLATION PARAMETERS FOR THRMAT ARRAY
                  BKP=FKP+1.
                  IF(BKP.GT.7.) BKP=7.
                  BNRG=4.*ALOG10(ENRG)-3.
C
                  RLOG=ALOG10(RR(N-1))
                  IF(RLOG.GE.FLXR(IRDIM)) THEN
                     BR=IRDIM
                     THRLOG=G3NTRP(THRMAT,IRDIM,NRGDIM,KPDIM,BR,BNRG,
     2                          BKP)+ALOG10(13./RR(N-1))
                  ELSE
C
                     DO 18 IR=1,IRDIM-1
                        IF(RLOG.LE.FLXR(IR+1)) THEN
                           BR=FLOAT(IR)+(RLOG-FLXR(IR))/(FLXR(IR+1)-
     2                                                  FLXR(IR))
                           GO TO 19
                        END IF
   18                CONTINUE
C
   19                CONTINUE
C
                     THRLOG=G3NTRP(THRMAT,IRDIM,NRGDIM,KPDIM,BR,BNRG,
     2                          BKP)
C
                  END IF
C
                  THRSHJ=10.**THRLOG
C
```

```
C
               ETATHR=THRSHJ*ALMDEL(IE)/7.392E-16/
     2                    SQRT(ABS(ALAM(IE)))/VVM(N-1)
C
C
C
             END IF
C
C
C  CALCULATE RHS OF D(ETA)/DT EQN
               WKMID=.5*(WEAK(N)+WEAK(N-1))
               STRMID=.5*(STRONG(N)+STRONG(N-1))
               DETDTW=AMAX1(0.,(ETANIL-ETAEQK))*WKMID
C
               DETDTS=AMAX1(0.,(ETANIL-ETATHR))*STPMID
C
C
C
               IF(DETDTW.EQ.0..AND.DETDTS.EQ.0.) THEN
                  ETANIL=ETANIL
                  RATETA=0.
                  GO TO 300
               END IF
C
               IF(DETDTW.GT.DETDTS) THEN
                  RATE=WKMID
                  ETAM=ETAEQK
                  RATETA=DETDTW
               ELSE
                  RATE=STRMID
                  ETAM=ETATHR
                  RATETA=DETDTS
               END IF
C
            IF(RATE.GT.10.) THEN
               WRITE(6,*) 'I,J,IE,WKMID,STRMID',I,J,IE,WKMID,STRMID
               WRITE(6,795) DETDTW,DETDTS,ETANIL
  795 FORMAT(1X,'DETDTW=',1PE12.3,2X,'DETDTS=',E12.3,2X,'ETA(N)=',E12.3)
               WRITE(6,799) N,ETAM,ETANIL,TM(N-1),TM(N),RATE
  799 FORMAT(1X,'N=',I6,2X,'ETAM=',1PE12.3,2X,'ETANIL=',E12.3,2X,
     2       'TM(N-1)=',F10.2,2X,'TM(N)=',F10.2,2X,'RATE=',E12.3)
               WRITE(6,*) 'STOPPING IN PPTM'
               STOP
            END IF
C
            ETAOLD=ETANIL
            ETANIL=ETAM+(ETAOLD-ETAM)*EXP(-(TM(N-1)-TM(N))*RATE)
C
C
C
C
C
  300 CONTINUE
C
C
C
         ETA(I,J,IE)=ETANIL
C
C
            BI=I
            BJ=J
            BT=TNORML(TSTART,TIMTAG,ITMMAX)
            VMVAL=G3NTRP(VM,IDIM,JDIM,ITMDIM,BI,BJ,BT)
            ENRG=ABS(ALAM(IE))*VMVAL
```

```
C
C
C
C  COMPUTE PARTICLE FLUX ASSOCIATED WITH ETA(I,J,IE).  IF COMPUTED
C   FLUX IS GREATER THAN UPPER LIMIT REFERENCE FLUX, THEN REDUCE ETA
C   TO BE CONSISTENT WITH REFERENCE FLUX.  REF:  SECTION 2.5.6 OF THE
C   FINAL REPORT FOR CONTRACT #F19628-87-K-0001.
C
                  IF(ISP.EQ.1) THEN
                     FLUX=7.392E-16*SQRT(ABS(ALAM(IE)))*VMVAL*
     2                    ETA(I,J,IE)/ALMDEL(IE)
                  ELSE
                     IF(ISP.EQ.2) THEN
                        AWT=1.
                     ELSE
                        AWT=16.
                     END IF
C
                     FLUX=1.731E-17*SQRT(ABS(ALAM(IE)))*VMVAL*
     2                    ETA(I,J,IE)/(ALMDEL(IE)*SQRT(AWT))
C
                  END IF
C
C
 892 FORMAT(1X,'VMVAL=',F8.3,2X,'ALMDEL=',1PE12.3,2X,'FLUX=',E12.3,2X,
     2       'FLXREF=',E12.3)
                  IF(FLUX.GT.FLXREF(I,J,IE)) THEN
C                    WRITE(6,*) 'COMPUTED FLUX EXCEEDS REFERENCE FLUX',
C    2                     ' I=',I,'  J=',J,'  IE=',IE,
C    3                     ' FLUX=',FLUX,' REF FLUX=',FLXREF(I,J,IE)
C  CORRECT ETA TO REFERENCE FLUX VALUE
                     ETAS=ETA(I,J,IE)
                     ETA(I,J,IE)=ETA(I,J,IE)*FLXREF(I,J,IE)/FLUX
C  DEBUG PRINTOUT
C                    WRITE(6,*) 'COMPUTED ETA= ',ETAS,
C    2                                    ' CORRECTED ETA=',ETA(I,J,IE)
                  END IF
C
C              END IF
C
C
C
C
          END IF
C
  23 CONTINUE
C
          IF(ISP.EQ.1) THEN
C
C  CALCULATE PRECIPITATING ENERGY FLUX AND FLUX SUMS
             BBI=I
             BBJ=J
             BIRM=G3NTRP(BIR,IDIM,JDIM,1,BBI,BBJ,1.)
             DETDTW=AMAX1(0.,(ETA(I,J,IE)-ETAEQK))*WEAK(1)
             DETDTS=AMAX1(0.,(ETA(I,J,IE)-ETATHR))*STRONG(1)
             RATETA=AMAX1(DETDTW,DETDTS)
             FLUXN=RATETA*BIRM/2.E13
             EFLUX(I,J,IE)=.8E-25*RATETA*BIRM*ABS(AALAM)*VVM(1)
             FLXSUM(I,J,IK)=FLXSUM(I,J,IK)+EFLUX(I,J,IE)
             DENSUM(I,J,IK)=DENSUM(I,J,IK)+FLUXN
C
          END IF
  24 CONTINUE
             ETA(I,1,IE)=ETA(I,JDIM-JWRAP+1,IE)
```

```
          ETA(I,2,IE)=ETA(I,JDIM-JWRAP+2,IE)
          ETA(I,JDIM,IE)=ETA(I,JWRAP,IE)
          IF(ISP.EQ.1) THEN
             EFLUX(I,1,IE)=EFLUX(I,JDIM-JWRAP+1,IE)
             EFLUX(I,2,IE)=EFLUX(I,JDIM-JWRAP+2,IE)
             EFLUX(I,JDIM,IE)=EFLUX(I,JWRAP,IE)
             FLXSUM(I,1,IK)=FLXSUM(I,JDIM-JWRAP+1,IK)
             FLXSUM(I,2,IK)=FLXSUM(I,JDIM-JWRAP+2,IK)
             FLXSUM(I,JDIM,IK)=FLXSUM(I,JWRAP,IK)
             DENSUM(I,1,IK)=DENSUM(I,JDIM-JWRAP+1,IK)
             DENSUM(I,2,IK)=DENSUM(I,JDIM-JWRAP+2,IK)
             DENSUM(I,JDIM,IK)=DENSUM(I,JWRAP,IK)
          END IF
C
C
C     PRINT THE TIME SPENT PROCESSING LATITUDE I
C
c     CALL DTIME(ITRCEN)
c     WRITE (LUERR,*) 'THE TIME SPENT TRACING PARTICLE ',IE,
c    $   'AT LATITUDE INDEX ',I,' WAS ',
c    $   ABS(ITRCEN-ITRCST)      ,' SECONDS.'
C
C
C
   20 CONTINUE
C
C
      WRITE(18,*) 'IE=',IE,'T=',TSTART
      WRITE(6,*) 'IE=',IE,'T=',TSTART
C
      IBM=IDIM
      DO 6 JJJ=1,JDIM
         IF(INT(BNDLOC(JJJ,LL)).LT.IBM) IBM=BNDLOC(JJJ,LL)
    6 CONTINUE
C
C
C     PRINT TOTAL COMP. TIME FOR THIS K VALUE
C
c        CALL DTIME(ITKEND)
c        WRITE (LUERR,*) 'THE TOTAL TIME SPENT TRACING K = ', IE,
c    $      ' WAS ', ABS(ITKEND-ITKST)      ,' SECONDS.'
C
   10 CONTINUE
C
C
C
C  CALCULATE PRECIPITATING FLUXES AND ENERGIES
C
C      INITIALIZATION OF PRECIPITATION ARRAYS
C
C
      DO 200 I=IBEG,ISTOP
      DO 200 J=JBEG,JSTOP
         IF(DENSUM(I,J,1).GT.0.) THEN
            EAVG(I,J,1)=FLXSUM(I,J,1)/DENSUM(I,J,1)
C  CONVERT EAVG TO EV FROM ERGS
            EAVG(I,J,1)=EAVG(I,J,1)*6.242E11
         ELSE
            EAVG(I,J,1)=0.
         END IF
C
C    1            EAVG(I,J,1)
C
  200 CONTINUE
```

```
C
C
C
C
      RETURN
      END
      SUBROUTINE BNDCHK(T,IE,BBI,BBJ,BIOLD,BBJOLD,TIMTAG,ETABND,BNDMAX,
     1             BNDLOC,JDIM,IEDIM,ITMDIM,ITMMAX,JWRAP,ETANIL,IEXIT)
C
C  VERSION 1.00                DATE: 01.11.88
C
C  PURPOSE:  SUBROUTINE TO DETERMINE WHETHER PARTICLE HAS CROSSED
C            BNDY OF CALCULATION.  IF SO, IEXIT IS SET EQUAL TO 2 AND
C            ETANIL=ETA VALUE AT BOUNDARY CROSSING PT IS CALCULATED.
C
C
C  INPUT:
C     T             TIME
C     IE            PARTICLE SPECIES (IE., ENERGY CHANNEL)
C     BBI           I LOCATION AT END OF TIME STEP
C     BBJ           J LOCATION AT END OF TIME STEP
C     BIOLD         I LOCATION AT BEGINNING OF TIME STEP
C     BBJOLD        J LOCATION AT BEGINNING OF TIME STEP
C     TIMTAG        VECTOR GIVING EVENT TIMES FOR WHICH DATA ARRAYS ARE
C                   GIVEN
C     ETABND        ETA DISTRIBUTION ON BOUNDARY (DEPENDS ON TIME)
C     BNDMAX        VECTOR GIVING MAX I VALUE OF BNDY LOCATION AT EACH
C                   J VALUE
C     BNDLOC        ARRAY GIVING BNDY LOCATION AS FUNCTION OF TIME
C     JDIM          NUMBER OF LONGITUDE GRID POINTS (INCL WRAPAROUND)
C     IEDIM         NUMBER OF ENERGY SPECIES
C     ITMDIM        MAX NUMBER OF TEMPORAL GRID POINTS
C     ITMMAX        NUMBER OF TEMPORAL PTS FILLED
C     JWRAP         J WRAP-AROUND
C
C  OUTPUT:
C     ETANIL        IF BNDY IS CROSSED, ETANIL IS THE VALUE OF ETA
C                   AT THE BNDY CROSSING PT
C     IEXIT         IF BNDY IS CROSSED, IEXIT= 2
C
C  PROGRAMMER: R. W. SPIRO
C
C
C
C
      DIMENSION TIMTAG(ITMDIM),BNDMAX(JDIM),BNDLOC(JDIM,ITMDIM)
      DIMENSION ETABND(JDIM,IEDIM,ITMDIM)
      COMMON/TMP6/III,JJJ
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C
C
C  IF SPATIAL BOUNDARY IS CROSSED, SET IEXIT=2, CALCULATE CROSSING
C    POINT, AND VALUE OF ETA AT CROSSING POINT
C
C  INITIAL CHECK - EVALUATE BI AT MAXIMUM BOUNDARY VALUES AT BJ=BBJ
C
         JJ=BBJ
         BIVAL=(BBJ-REAL(JJ))*BNDMAX(JJ+1)+(REAL(JJ+1)-BBJ)*BNDMAX(JJ)
C
         IF(BBI.LE.BIVAL) THEN
```

```
C                             WE NEED TO PERFORM TIME INTERPOLATION OF
C                             BOUNDARY LOCATION AND CHECK AGAINST
C                             LOCATION (BBI,BBJ)
C
            BBT=TNORML(T,TIMTAG,ITMMAX)
            ITM=BBT
            BNDJ=(1.-BBT+REAL(ITM))*BNDLOC(JJ,ITM)
     1          +(BBT-REAL(ITM))*BNDLOC(JJ,ITM+1)
            BNDJP=(1.-BBT+REAL(ITM))*BNDLOC(JJ+1,ITM)
     1          +(BBT-REAL(ITM))*BNDLOC(JJ+1,ITM+1)
C
            BIVAL=(BBJ-REAL(JJ))*BNDJP+(REAL(JJ+1)-BBJ)*BNDJ
C
            IF(BBI.LE.BIVAL) THEN
C
C  PARTICLE HAS CROSSED SPATIAL BOUNDARY
C    SET IEXIT=2 AND CALCULATE CROSSING POINT
C
              IEXIT=2
C
            BJOLD=BBJOLD
C  NEED TO GET BBJ AND BJOLD IN SAME MODULUS
            IF(BBJOLD-BBJ.GT.REAL(JDIM-JWRAP)/2.) THEN
              BJOLD=BBJOLD-REAL(JDIM-JWRAP)
            END IF
C
            IF(BBJ-BBJOLD.GT.REAL(JDIM-JWRAP)/2.) THEN
              BJOLD=BJOLD+REAL(JDIM-JWRAP)
            END IF
C
C  LINE1 IS LINE FROM OLD LOCATION (BIOLD,BJOLD) TO NEW LOCATION
C    (BBI,BBJ).  DESCRIBE LINE AS  BI=DIDJ*BJ+BICON
C
              IF(ABS(BBJ-BJOLD).LE.1.E-6) THEN
                BJCRSS=BBJ
              ELSE
                DIDJ1=(BBI-BIOLD)/(BBJ-BJOLD)
                BICON1=(BIOLD*BBJ-BBI*BJOLD)/(BBJ-BJOLD)
                JJBEG=MIN(BJOLD,BBJ)
                JJEND=MAX(BJOLD,BBJ)
C
C  CHECK ALL POSSIBLE CROSSING SEGMENTS
C
                DO 40 JJ=JJBEG,JJEND
                  BNDJJ=(1.-BBT+REAL(ITM))*BNDLOC(JJ,ITM)
     1                +(BBT-REAL(ITM))*BNDLOC(JJ,ITM+1)
                  BNDJJP=(1.-BBT+REAL(ITM))*BNDLOC(JJ+1,ITM)
     1                +(BBT-REAL(ITM))*BNDLOC(JJ+1,ITM+1)
C
                  DIDJ2=(BNDJJP-BNDJJ)
                  BICON2=BNDJJ*REAL(JJ+1)-BNDJJP*REAL(JJ)
                  IF(ABS(DIDJ2-DIDJ1).LE.1.E-6) GO TO 40
C
                  BJCRSS=(BICON1-BICON2)/(DIDJ2-DIDJ1)
C
C  CHECK IF CROSSING IS BETWEEN JJ AND JJ+1
C
                  IF(BJCRSS.GE.REAL(JJ).AND.BJCRSS.LE.REAL(JJ+1))THEN
C                     BOUNCE OUT OF DO 40 LOOP
                     GO TO 50
                  END IF
                  IF(JJBEG.EQ.JJEND) GO TO 50
   40           CONTINUE
C
```

```
C  DEBUG - FAILED TO FIND BOUNDARY CROSSING
C
C     WRITE(6,*) 'FAILED TO FIND BNDY CROSSING IN SUBROUTINE PPTM'
C     WRITE(6,*) 'T,IE,BBI,BBJ,BIOLD,BBJOLD',T,IE,BBI,BBJ,BIOLD,BBJOLD
C     WRITE(6,*) 'BJCRSS',BJCRSS
      BJCRSS=BBJ
C     WRITE(6,*) 'USING BBJ AS CROSSING POINT=',BBJ
C
C
   50              CONTINUE
C
C  CROSSING POINT IS BJCRSS.  COMPUTE ETANIL= ETA AT BNDY CROSSING
C
                   JJ=BJCRSS
                   ETAJ=(1.-BBT+REAL(ITM))*ETABND(JJ,IE,ITM)
     1                  +(BBT-REAL(ITM))*ETABND(JJ,IE,ITM+1)
                   ETAJP=(1.-BBT+REAL(ITM))*ETABND(JJ+1,IE,ITM)
     1                  +(BBT-REAL(ITM))*ETABND(JJ+1,IE,ITM+1)
C
                   ETANIL=(1.-BJCRSS+REAL(JJ))*ETAJ+(BJCRSS-REAL(JJ))*ETAJP
C                  IF(IE.EQ.1.OR.IE.EQ.17) WRITE(6,900)III,JJJ,ITM,IE,
C    1             BBT,BBI,BBJ,JJ,ETAJ,ETAJP,ETANIL
C900               FORMAT(1H ,'III,JJJ,ITM,IE,BBT,BBI,BBJ,JJ,ETAJ,ETAJP,
C    1             ETANIL'/4I4,3F15.5,I4,3E15.5)
                END IF
C
            END IF
C
      END IF
C
      RETURN
      END
      FUNCTION DVEFDI(I,J,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)
C
C   VERSION 1.00                               DATE: 12.16.87
C           1.01A                                    02.02.89
C
C   PURPOSE:  FUNCTION SUBPROGRAM TO COMPUTE THE I DERIVATIVE OF THE
C             EFFECTIVE POTENTIAL AT GRID POINT (I,J) AT NORMALIZED
C             TIME BT.
C
C   INPUT:
C      I                      I COORDINATE OF THE GRID POINT
C      J                      J COORDINATE OF THE GRID POINT
C      BT                     NORMALIZED TIME
C      AALAM                  ENERGY INVARIANT (EV (RE/NT)**2/3)
C      V                      ELECTRIC POTENTIAL ARRAY (VOLTS)
C      VM                     (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
C      IDIM                   I DIMENSION OF V AND VM
C      JDIM                   J DIMENSION OF V AND VM
C      ITMDIM                 TIME DIMENSION OF V AND VM
C
C   OUTPUT:
C      DVEFDI                 D(V+AALAM*VM)/DI AT (I,J,BT)
C
C   PROGRAMMER:  R. W. SPIRO
C
C
C
      COMMON /GRID/DLAM,DPSI,RI,JWRAP
C
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
```

```
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM)
C
      ITT=BT
      FT=REAL(ITT)
      IMIN=1
      IMAX=IDIM
      IF(J.LT.2) THEN
         JX=J+(JDIM-JWRAP)
      ELSE IF(J.GT.JDIM-1) THEN
         JX=J-(JDIM-JWRAP)
      ELSE
         JX=J
      END IF
C
C   COMPUTE DVEFDI
C
      IF(I.EQ.IMIN) THEN
         DVEFDI=(1.-BT+FT)*(-1.5*(V(IMIN,JX,ITT)+AALAM*VM(IMIN,JX,ITT))
     1           +2.*(V(IMIN+1,JX,ITT)+AALAM*VM(IMIN+1,JX,ITT))
     2           -.5*(V(IMIN+2,JX,ITT)+AALAM*VM(IMIN+2,JX,ITT)))
     3         +(BT-FT)*(-1.5*(V(IMIN,JX,ITT+1)+AALAM*VM(IMIN,JX,ITT+1))
     4           +2.*(V(IMIN+1,JX,ITT+1)+AALAM*VM(IMIN+1,JX,ITT+1))
     5           -.5*(V(IMIN+2,JX,ITT+1)+AALAM*VM(IMIN+2,JX,ITT+1)))
      ELSE IF(I.EQ.IMAX) THEN
         DVEFDI=(1.-BT+FT)*(-1.5*(V(IMAX,JX,ITT)+AALAM*VM(IMAX,JX,ITT))
     1           +2.*(V(IMAX-1,JX,ITT)+AALAM*VM(IMAX-1,JX,ITT))
     2           -.5*(V(IMAX-2,JX,ITT)+AALAM*VM(IMAX-2,JX,ITT)))
     3         +(BT-FT)*(-1.5*(V(IMAX,JX,ITT+1)+AALAM*VM(IMAX,JX,ITT+1))
     4           +2.*(V(IMAX-1,JX,ITT+1)+AALAM*VM(IMAX-1,JX,ITT+1))
     5           -.5*(V(IMAX-2,JX,ITT+1)+AALAM*VM(IMAX-2,JX,ITT+1)))
      ELSE IF (I.GT.IMIN.AND.I.LT.IMAX) THEN
         DVEFDI=((1.-BT+FT)*(V(I+1,JX,ITT)+AALAM*VM(I+1,JX,ITT)
     1           -V(I-1,JX,ITT)-AALAM*VM(I-1,JX,ITT))+
     2           (BT-FT)*(V(I+1,JX,ITT+1)+AALAM*VM(I+1,JX,ITT+1)
     3           -V(I-1,JX,ITT+1)-AALAM*VM(I-1,JX,ITT+1)))/2.
      ELSE
         WRITE (LUERR,*) 'Error in Function DVEFDI, I out of range.'
         WRITE (LUERR,*) 'I = ',I,' JX = ',JX,' ITT = ',ITT
         WRITE (LUERR,*) ' '
      END IF
C
      RETURN
      END
      FUNCTION DVEFDJ(I,J,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)
C
C   VERSION 1.00                            DATE: 12.16.87
C           1.01A                                 02.02.89
C
C   PURPOSE:  FUNCTION SUBPROGRAM TO COMPUTE THE J DERIVATIVE OF THE
C             EFFECTIVE POTENTIAL AT GRID POINT (I,J) AT NORMALIZED
C             TIME BT.
C
C   INPUT:
C      I                    I COORDINATE OF THE GRID POINT
C      J                    J COORDINATE OF THE GRID POINT
C      BT                   NORMALIZED TIME
C      AALAM                ENERGY INVARIANT (EV (RE/NT)**2/3)
C      V                    ELECTRIC POTENTIAL ARRAY (VOLTS)
C      VM                   (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
C      IDIM                 I DIMENSION OF V AND VM
C      JDIM                 J DIMENSION OF V AND VM
C      ITMDIM               TIME DIMENSION OF V AND VM
C      JWRAP                J WRAP AROUND
```

```
C
C   OUTPUT:
C      DVEFDI                  D(V+AALAM*VM)/DI AT (I,J,BT)
C
C   PROGRAMMER:  R. W. SPIRO
C
C
C
      COMMON /GRID/DLAM,DPSI,RI,JWRAP
C
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1       LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2       LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3       LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM)
C
      ITT=BT
      FT=REAL(ITT)
C
C   PUT J IN CORRECT MODULUS FOR CALCULATION
C
      IF(J.LT.2) THEN
         JX=J+(JDIM-JWRAP)
      ELSE IF(J.GT.JDIM-1) THEN
         JX=J-(JDIM-JWRAP)
      ELSE
         JX=J
      END IF
C
C   COMPUTE DVEFDJ
C
      DVEFDJ=((1.-BT+FT)*(V(I,JX+1,ITT)+AALAM*VM(I,JX+1,ITT)
     1         -V(I,JX-1,ITT)-AALAM*VM(I,JX-1,ITT))+
     2         (BT-FT)*(V(I,JX+1,ITT+1)+AALAM*VM(I,JX+1,ITT+1)
     3         -V(I,JX-1,ITT+1)-AALAM*VM(I,JX-1,ITT+1)))/2.
C
      RETURN
      END
      FUNCTION G3NTRP(A,IMAX,JMAX,KMAX,BI,BJ,BK)
C
C  VERSION 1.00                        DATE: 01.11.88
C          1.01A                             02.02.89

C
C  PURPOSE: FUNCTION SUBPROGRAM TO PERFORM A GENERAL 3-D LINEAR
C           INTERPOLATION OF ARRAY A(I,J,K) AT PT(BV(1),BV(2),BV(3))
C
C  INPUT:
C      A          3-D ARRAY TO BE INTERPOLATED
C      IMAX       I DIMENSION OF ARRAY A
C      JMAX       J DIMENSION OF ARRAY A
C      KMAX       K DIMENSION OF ARRAY A
C      BI         FLOATING POINT VALUE TO INTERPOLATE IN I DIMENSION
C      BJ         FLOATING POINT VALUE TO INTERPOLATE IN J DIMENSION
C      BK         FLOATING POINT VALUE TO INTERPOLATE IN K DIMENSION
C
C
C  OUTPUT:
C      G3NTRP     INTERPOLATED VALUES OF ARRAY A
C
C                             INTERPOLATE A
C
C
C
```

```
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION A(IMAX,JMAX,KMAX)
      DIMENSION NDX(3),NDIM(3),BV(3),COEF(3,2)
C
C PREPARE INDICES FOR INTERPOLATION
C
      NDIM(1)=IMAX
      NDIM(2)=JMAX
      NDIM(3)=KMAX
      BV(1)=BI
      BV(2)=BJ
      BV(3)=BK
      DO 10 L=1,3
         NDX(L)=BV(L)
         IF(NDX(L).LT.1) NDX(L)=1
         IF(NDX(L).GT.NDIM(L)-1) NDX(L)=NDIM(L)-1
         IF(NDX(L).LE.0) NDX(L)=1
C
         FNDX=REAL(NDX(L))
         COEF(L,1)=1.-BV(L)+FNDX
         COEF(L,2)=BV(L)-FNDX
   10 CONTINUE
C
C
      G3NTRP=0.
      KSTOP = MIN(KMAX,2)
      JSTOP = MIN(JMAX,2)
      DO 20 I=1,2
      DO 20 J=1,JSTOP
      DO 20 K=1,KSTOP
         G3NTRP=G3NTRP+
     1         COEF(1,I)*COEF(2,J)*COEF(3,K)*A(NDX(1)+I-1,NDX(2)+J-1,
     2                                       NDX(3)+K-1)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE MODFIX(BBJ,JDIM,JWRAP)
C
C VERSION 1.00                        DATE: 01.15.88
C
C PURPOSE:  SUBROUTINE TO ADJUST BBJ TO BE BETWEEN 2 AND JDIM-1
C
C INPUT:
C     BBJ           VALUE TO BE ADJUSTED
C     JDIM          J DIMENSION OF ARRAYS
C     JWRAP         J WRAP AROUND
C
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX, JEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      IF(BBJ.LT.REAL(2)) BBJ=BBJ+REAL(JDIM-JWRAP)
      IF(BBJ.GT.REAL(JDIM-1)) BBJ=BBJ-REAL(JDIM-JWRAP)
      RETURN
      END
      SUBROUTINE MOVER(T,DTTRY,BIOLD,BJOLD,EPS,BBISCL,BBJSCL,DTUSED,
     1 DTNEXT,AALAM,V,VM,TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM,BBI,BBJ)
C
C VERSION 1.00                   DATE: 01.15.88
C
C PURPOSE:  SUBROUTINE TO ADVANCE ONE PARTICLE LOCATION ONE TIME
```

```
C  TAKE TWO HALF STEPS
C
      DT2=DT/2.
      CALL RK4(TSAV,DT2,BBISAV,BBJSAV,BBIOUT,BBJOUT,AALAM,V,VM,
     1         TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C        SECOND HALF STEP
C
      BBIIN=BBIOUT
      BBJIN=BBJOUT
      CALL RK4(T+DT2,DT2,BBIIN,BBJIN,BBIA,BBJA,AALAM,V,VM,
     1         TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C  TAKE LARGE STEP
C
      CALL RK4(TSAV,DT,BBISAV,BBJSAV,BBIB,BBJB,AALAM,V,VM,
     1         TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C  EVALUATE ACCURACY
C
      ERRMAX=0.
C
      BIERR=BBIB-BBIA
      ERRMAX=MAX(ERRMAX,ABS(BIERR/BBISCL))
C
      BJERR=BBJB-BBJA
      ERRMAX=MAX(ERRMAX,ABS(BJERR/BBJSCL))
C
C  SCALE RELATIVE TO REQUIRED TOLERANCE
C
      ERRMAX=ERRMAX/EPS
C
      IF(ERRMAX.GT.ONE) THEN
C                              ERROR TOO LARGE, REDUCE STEPSIZE
         DT=SAFETY*DT*(ERRMAX**PSHRNK)
         GO TO 10
C                 FOR ANOTHER TRY
      ELSE
         DTUSED=DT
         IF(ERRMAX.GT.ERRCON) THEN
            DTNEXT=SAFETY*DT*(ERRMAX**PGROW)
         ELSE
            DTNEXT=4.*DT
         END IF
      END IF
C
C  PUT IN FIFTH ORDER CORRECTION
C
      BBI=BBIA+BIERR*FCOR
         IF(BBI.GT.REAL(IDIM)) BBI=REAL(IDIM)
      BBJ=BBJA+BJERR*FCOR
C
C
      RETURN
      END
      SUBROUTINE RK4(T,DT,BBIIN,BBJIN,BBIOUT,BBJOUT,AALAM,V,VM,TIMTAG,
     1                ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C   VERSION 1.00                DATE: 12.30.87
C
C  PURPOSE:  SUBROUTINE TO ADVANCE ONE PARTICLE LOCATION ONE TIME
C            STEP DT USING 4TH ORDER RUNGE KUTTA ALGORITHM.
C
C  INPUT:
```

```
C       T               TIME PARAMETER
C       DT              TIME STEP, DT.LT.0 MOVES PARTICLES BACK IN TIME
C                                       DT.GT.0 MOVES PARTICLES FORWARD
C       BBIIN           CURRENT I LOCATION OF PARTICLE
C       BBJIN           CURRENT J LOCATION OF PARTICLE
C       AALAM           ENERGY INVARIANT OF PARTICLE
C       V               ELECTRIC POTENTIAL ARRAY (VOLTS)
C       VM              (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
C       TIMTAG          VECTOR GIVING UNIVERSAL TIMES FOR WHICH DATA ARRAYS
C                            ARE GIVEN
C       ALPHA           LATITUDINAL GRID SPACING VECTOR
C       BETA            LOCAL TIME GRID SPACING VECTOR
C       BIR             IONSOPHERIC MAGNETIC FIELD STRENGTH
C       IDIM            NUMBER OF LATITUDINAL GRID POINTS
C       JDIM            NUMBER OF LOCAL TIME GRID POINTS + WRAPAROUND
C       ITMDIM          MAX NUMBER OF TEMPORAL GRID POINTS
C
C   OUTPUT:
C       BBIOUT          COMPUTED I LOCATION AT END OF TIME STEP
C       BBJOUT          COMPUTED J LOCATION AT END OF TIME STEP
C
C   PROGRAMMER: R. W. SPIRO
C
C   REFERENCE:  ABRAMOWITZ AND STEGUN, HANDBOOK OF MATHEMATICAL FUNCTIONS
C               NATIONAL BUREAU OF STANDARDS, 1970. P. 896, SECT. 25.5.10
C
C
      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1        LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2        LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3        LUCOLT,LUBNDY,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION ALPHA(IDIM), BETA(IDIM), BIR(IDIM,JDIM)
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM),TIMTAG(ITMDIM)
      DIMENSION BITEMP(4), BJTEMP(4)
C
C
C    EVALUATE DERIVATIVE AT TIME=T, LOCATION=(BBI,BBJ)
      BT=TNORML(T,TIMTAG,ITMDIM)
      BIUSE=BBIIN
      BJUSE=BBJIN
      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,
     1             ITMDIM,DBIDT,DBJDT)
      BITEMP(1)=DT*DBIDT
      BJTEMP(1)=DT*DBJDT
C
C    EVALUATE DERIVATIVE AT TIME=T+DT/2, LOCATION=(BBIIN+BITEMP(1)/2,
C                                                  BBJIN+BJTEMP(1)/2)
      BT=TNORML(T+DT/2.,TIMTAG,ITMDIM)
      BIUSE=BBIIN+.5*BITEMP(1)
      BJUSE=BBJIN+.5*BJTEMP(1)
      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,
     1             ITMDIM,DBIDT,DBJDT)
      BITEMP(2)=DT*DBIDT
      BJTEMP(2)=DT*DBJDT
C
C    EVALUATE DERIVATIVE AT TIME=T+DT/2, LOCATION=(BBIIN+BITEMP(2)/2,
C                                                  BBJIN+BJTEMP(2)/2)
      BIUSE=BBIIN+.5*BITEMP(2)
      BJUSE=BBJIN+.5*BJTEMP(2)
      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,
     1             ITMDIM,DBIDT,DBJDT)
      BITEMP(3)=DT*DBIDT
```

```
      BJTEMP(3)=DT*DBJDT                                                MSM37770
C                                                                       MSM37780
C   EVALUATE DERIVATIVE AT TIME=T+DT, LOCATION=(BBIIN+BITEMP(3),        MSM37790
C                                              BBJIN+BJTEMP(3))         MSM37800
      BT=TNORML(T+DT,TIMTAG,ITMDIM)                                     MSM37810
      BIUSE=BBIIN+BITEMP(3)                                             MSM37820
      BJUSE=BBJIN+BJTEMP(3)                                             MSM37830
      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,   MSM37840
     1            ITMDIM,DBIDT,DBJDT)                                   MSM37850
      BITEMP(4)=DT*DBIDT                                                MSM37860
      BJTEMP(4)=DT*DBJDT                                                MSM37870
C                                                                       MSM37880
C                                                                       MSM37890
C   COMPUTE NEW LOCATIONS AT END OF TIME STEP                           MSM37900
      BBIOUT=BBIIN+(BITEMP(1)+2.*(BITEMP(2)+BITEMP(3))+BITEMP(4))/6.    MSM37910
      BBJOUT=BBJIN+(BJTEMP(1)+2.*(BJTEMP(2)+BJTEMP(3))+BJTEMP(4))/6.    MSM37920
      RETURN                                                            MSM37930
      END                                                               MSM37940
      SUBROUTINE TCHK(T,TSTOP,DTNOM,BBI,BBJ,IEUSE,ETABEG,IDIM,JDIM,     MSM37950
     1                IEDIM,ETANIL,IEXIT)                               MSM37960
C                                                                       MSM37970
C  VERSION 1.00                 DATE: 01.11.88                          MSM37980
C                                                                       MSM37990
C  PURPOSE:  SUBROUTINE TO CHECK WHETHER TIME BOUNDARY WAS CROSSED THIS MSM38000
C            TIME STEP.  IS SO, SET IEXIT=1 AND INTERPOLATE TO FIND     MSM38010
C            ETABEG AT LOCATION (BBI,BBJ).                              MSM38020
C                                                                       MSM38030
C                                                                       MSM38040
C                                                                       MSM38050
C  INPUT:                                                               MSM38060
C     T             TIME                                                MSM38070
C     TSTOP         TIME AT WHICH TO END PARTICLE TRACEBACK             MSM38080
C     DTNOM         NOMINAL TIME STEP TO BEGIN TRACES.                  MSM38090
C     BBI           I LOCATION OF PARTICLE                              MSM38100
C     BBJ           J LOCATION OF PARTICLE                              MSM38110
C     IEUSE         ENERGY INDEX OF PARTICLS                            MSM38120
C     ETABEG        INITIAL ETA DISTRIBUTION (IE., AT T=TSTOP)          MSM38130
C     IDIM          NUMBER OF LATITUDINAL GRID POINTS                   MSM38140
C     JDIM          NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)  MSM38150
C     IEDIM         NUMBER OF ENERGY SPECIES                            MSM38160
C                                                                       MSM38170
C  OUTPUT:                                                              MSM38180
C     ETANIL        ETABEG VALUE AT LOCATION (BBI,BBJ) AT TIME T FOR    MSM38190
C                   ENERGY SPECIES IEUSE IF PARTICLE HAS CROSSED        MSM38200
C                   TIME BOUNDARY                                       MSM38210
C     IEXIT         SET EQUAL TO 1 IF PARTICLE HAS CROSSED TIME BNDY    MSM38220
C                                                                       MSM38230
C  PROGRAMMER: R. W. SPIRO                                              MSM38240
C                                                                       MSM38250
C                                                                       MSM38260
C                                                                       MSM38270
C                                                                       MSM38280
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,        MSM38290
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,       MSM38300
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,    MSM38310
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH         MSM38320
      DIMENSION ETABEG(IDIM,JDIM,IEDIM)                                 MSM38330
C                                                                       MSM38340
C  IF TIME BOUNDARY IS CROSSED, SET IEXIT=1 AND INTERPOLATE             MSM38350
C    TO FIND ETABEG AT LOCATION (BBI,BBJ)                               MSM38360
C                                                                       MSM38370
         IF((ABS(T-TSTOP).LT..01*ABS(DTNOM)).OR.                        MSM38380
     1      (DTNOM.GT.0..AND.T.GE.TSTOP).OR.                            MSM38390
     2      (DTNOM.LT.0..AND.T.LE.TSTOP)) THEN                          MSM38400
```

```
          IEXIT=1
          RIEUSE=REAL(IEUSE)
          ETANIL=G3NTRP(ETABEG,IDIM,JDIM,IEDIM,BBI,BBJ,RIEUSE)
        END IF
C
      RETURN
      END
      FUNCTION TNORML(TT,TIMTAG,ITMMAX)
C
C   VERSION 1.00                              DATE: 12.16.87
C
C   PURPOSE:  FUNCTION TO CALCULATE RUN-NORMALIZED TIME CORRESPONDING
C             TO TIME TT
C
C   INPUT:
C     TT                    TIME IN SECONDS
C     TIMTAG                VECTOR GIVING TIMES FOR WHICH MODEL ARRAYS
C                           ARE CALCULATED
C     ITMMAX                NUMBER OF TEMPORAL GRID POINTS USED
C
C   OUTPUT:
C     TNORML                NON-INTEGER TIME INDEX CORRESPONDING TO
C                           TIME TT
C
C   PROGRAMMER:  R. W. SPIRO
C
C
C
        COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION TIMTAG(ITMMAX)
C
      DO 10 I=1,ITMMAX-1
        IF((TT.GE.TIMTAG(I).AND.TT.LE.TIMTAG(I+1)).OR.
     1     (TT.LE.TIMTAG(I).AND.TT.GE.TIMTAG(I+1))) THEN
          II=I
          GO TO 20
        END IF
   10 CONTINUE
C
C   TIME TT NOT WITHIN TIMTAG RANGE
C
      WRITE(LUERR,*)'TT NOT WITHIN TIMTAG RANGE, TT= ',TT
      WRITE(LUERR,*)'TIMTAG(1)=',TIMTAG(1)
      WRITE(LUERR,*)'ITMMAX=',ITMMAX
      WRITE(LUERR,*)'TIMTAG(ITMMAX)=',TIMTAG(ITMMAX)
      WRITE(LUERR,*) 'STOPPING PROGRAM IN FUCTION TNORML'
      STOP
C
C   LOCALIZED TT TO WITHIN THE RIGHT TIME RANGE
C
   20 CONTINUE
      TNORML=REAL(II)+(TT-TIMTAG(II))/(TIMTAG(II+1)-TIMTAG(II))
C
      RETURN
      END
      SUBROUTINE VLOCTY(BBII,BBJJ,BT,AALAM,ALPHA,BETA,BIR,V,VM,
     1                  IDIM,JDIM,ITMDIM,DBIDT,DBJDT)
C
C   VERSION 1.00                              DATE: 12.16.87
C           1.01A                                   02.02.89
C
```

```
C   PURPOSE:  SUBROUTINE TO COMPUTE I AND J COMPONENTS OF PARTICLE
C             VELOCITY WITH ENERGY INVARIANT AALAM AT LOCATION
C             (BBI,BBJ) AT NORMALIZED TIME BT.
C
C   INPUT:
C      BBII                    I LOCATION OF PARTICLE
C      BBJJ                    J LOCATION OF PARTICLE
C      BT                      NORMALIZED TIME
C      AALAM                   ENERGY INVARIANT (EV (RE/NT)**2/3)
C      ALPHA                   LATITUDINAL GRID SPACING VECTOR
C      BETA                    LONGITUDINAL GRID SPACING VECTOR
C      BIR                     IONOSPHERIC RADIAL MAGNETIC FIELD STRENGTH
C      V                       ELECTRIC POTENTIAL ARRAY (VOLTS)
C      VM                      (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
C      IDIM                    1ST (LAT) DIMENSION OF V, VM, AND BIR
C      JDIM                    2ND (LONG) DIMENSION OF V,VM, AND BIR
C      ITMDIM                  3RD (TIME) DIMENSION OF V AND VM
C
C   OUTPUT:
C      DBIDT                   I (LAT) VELOCITY OF PARTICLE (I UNITS/SEC)
C      DBJDT                   J (LAT) VELOCITY OF PARTICLE (J UNITS/SEC)
C
C   PROGRAMMER:  R. W  SPIRO
C
C
C
      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION ALPHA(IDIM),BETA(IDIM),BIR(IDIM,JDIM)
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM)
      DIMENSION DVDI2(2,2),DVDJ2(2,2)
C
      BBI=BBII
      BBJ=BBJJ
      CALL MODFIX(BBJ,JDIM,JWRAP)
C   INTERPOLATE GRID ARRAY FOR LOCATION (BBI,BBJ)
C
      BIR2=G3NTRP(BIR,IDIM,JDIM,1,BBI,BBJ,1.)
      ALPH2=G3NTRP(ALPHA,IDIM,1,1,BBI,1.,1.)
      BETA2=G3NTRP(BETA,IDIM,1,1,BBI,1.,1.)
C
C   COMPUTE DVEFDI AND DVEFDJ AT NORMALIZED TIME BT FOR GRID PTS
C     SURROUNDING (BBI,BBJ)
C
      I=BBI
      FI=REAL(I)
      J=BBJ
      FJ=REAL(J)
C
      DO 10 II=1,2
         IUSE=I+II-1
         IF(IUSE.GT.IDIM) IUSE=IDIM
      DO 20 JJ=1,2
         JUSE=J+JJ-1
         DVDI2(II,JJ)=DVEFDI(IUSE,JUSE,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)
         DVDJ2(II,JJ)=DVEFDJ(IUSE,JUSE,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)
   20 CONTINUE
   10 CONTINUE
C
C   INTERPOLATE DVDI2 AND DVDJ2 FOR LOCATION (BBI,BBJ)
```

```
C
      DVDI=(1.-BBJ+FJ)*((1.-BBI+FI)*DVDI2(1,1)+(BBI-FI)*DVDI2(2,1))
     1        + (BBJ-FJ)*((1.-BBI+FI)*DVDI2(1,2)+(BBI-FI)*DVDI2(2,2))
C
      DVDJ=(1.-BBJ+FJ)*((1.-BBI+FI)*DVDJ2(1,1)+(BBI-FI)*DVDJ2(2,1))
     1        + (BBJ-FJ)*((1.-BBI+FI)*DVDJ2(1,2)+(BBI-FI)*DVDJ2(2,2))
C
C   COMPUTE VELOCITIES
C
      DBIDT=DVDJ*1000./(BIR2*ALPH2*BETA2*DLAM*DPSI*RI**2)
      DBJDT=-DVDI*1000./(BIR2*ALPH2*BETA2*DLAM*DPSI*RI**2)
C
      RETURN
      END
      SUBROUTINE RDGRID(IDIM,JDIM,ALPHA,BETA,TETA,PHI,COLAT,BIR,SINI)
C
C  VERSION 1.0                         DATE: 01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE:  READ GRID SYSTEM COORDINATES AND CALCULATE ESSENTIAL GRID
C                QUANTITIES
C
      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1       LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2       LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3       LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION      ALPHA(IDIM),BETA(IDIM),TETA(IDIM),PHI(JDIM)
      DIMENSION COLAT(IDIM,JDIM),BIR(IDIM,JDIM),SINI(IDIM,JDIM)
C
      CHARACTER   ID(2)
      PI=ATAN2(0.,-1.)
      JWRAP = 3
C
C  READ GRID PARAMETERS
C
      OPEN(UNIT=LUCORD,FILE='COORD',STATUS='OLD')
C
      READ(LUCORD,1000) ID,OFFSET,DLAM,DPSI,RI,RE
1000  FORMAT(2A4,1X,5(E12.6,1X))
      READ(LUCORD,1001)TETA,ALPHA,BETA,PHI
1001  FORMAT(10(E12.6,1X))
C
      DO 10 I=1,IDIM
      DO 20 J=1,JDIM
         COSTET=COS(TETA(I))*COS(OFFSET)+SIN(TETA(I))*COS(PHI(J))*
     1             SIN(OFFSET)
         BIR(I,J)=62000.*COSTET
         COLAT(I,J)=ACOS(COSTET)
         SINI(I,J)=2.*COSTET/SQRT(1.+3.*COSTET**2)
   20 CONTINUE
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE MLTSET(ISEC,IDIM,JDIM,ALOCT)
C
C  VERSION 1.0                         DATE: 01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE: CALCULATE TIME DEPENDENT ALOCT OF ROTATING GRID PTS
```

```
C
C  INPUTS:
C      ISEC         TIME (SECONDS)
C      IDIM         I DIMENSION
C      JDIM         J DIMENSION
C
C  OUTPUT:
C      ALOCT        LOCAL TIME ARRAY CORRESPONDING TO T=ISEC
C
      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
      DIMENSION ALOCT(IDIM,JDIM)
C
      PI=ATAN2(0.,-1.)
      DPHI=2.*PI/REAL(JDIM-JWRAP)
      DPHIDT=2.*PI/86400.
      T=REAL(ISEC)
C
      DO 10 I=1,IDIM
      DO 10 J=1,JDIM
         ALOCT(I,J)=REAL(J-3)*DPHI+T*DPHIDT
         IF(ALOCT(I,J).GT.2.*PI) ALOCT(I,J)=ALOCT(I,J)-2.*PI
         IF(ALOCT(I,J).LT.0.) ALOCT(I,J)=ALOCT(I,J)+2.*PI
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE VMLSET(ITMCUR,VM,BIR,SINI,IDIM,JDIM,ITMDIM,VMLOSS)
C
C  VERSION 1.0                          DATE:  01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE:  COMPUTE VMLOSS ARRAY FOR ITMCUR
C
      DIMENSION VMLOSS(IDIM,JDIM,ITMDIM),BIR(IDIM,JDIM),SINI(IDIM,JDIM)
      DIMENSION VM(IDIM,JDIM,ITMDIM)
C
      DO 10 I=1,IDIM
      DO 10 J=1,JDIM
         VMLOSS(I,J,ITMCUR)=(SINI(I,J)/BIR(I,J))*VM(I,J,ITMCUR)**2
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE FNDBND(ITMCUR,COLAT,ALOCT,A2,B2,DX2,DY2,IDIM,JDIM,
     1                     ITMDIM,BNDLOC)
C
C  VERSION 1.0                       DATE: 01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE:  COMPUTE TIME DEPENDENT OUTER BOUNDARY LOCATION
C
        COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION COLAT(IDIM,JDIM),ALOCT(IDIM,JDIM)
      DIMENSION BNDLOC(JDIM,ITMDIM)
C
      PI = ATAN2(0.,-1.)
      DO 10 J=1,JDIM
      DO 20 I=1,IDIM-1
```

```
         IUSE=I
         THETAB=THET(A2,B2,DX2,DY2,ALOCT(I+1,J))*PI/180.
         IF(COLAT(I+1,J).GE.THETAB) GO TO 30
   20 CONTINUE
C
C  ERROR EXIT
      WRITE(LUERR,*) 'ERROR IN FFNDBND'
      WRITE(LUERR,*) 'THETAB=',THETAB,'J=',J,'I=',I
      WRITE(LUERR,*) 'STOPPING PGM'
      STOP
C
   30 CONTINUE
C
      THETAC=THET(A2,B2,DX2,DY2,ALOCT(IUSE,J))*PI/180.
      BNDLOC(J,ITMCUR)=(REAL(IUSE)*(COLAT(IUSE+1,J)-THETAB)+
     1                  REAL(IUSE+1)*(THETAC-COLAT(IUSE,J)))/
     2                  (COLAT(IUSE+1,J)-COLAT(IUSE,J)-THETAB+THETAC)
C
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE OUTP(R,ISIZE,JSIZE,IBEG,IEND,IINC,JBEG,JEND,JINC,
     $                XSCALE,TITLE,NTP,NCOL)
C
C AUTHOR: R.W. SPIRO        LAST MODIFIED: 06-25-85
C
C*********************************************************************
C DESCRIPTION OF INPUT PARAMETERS
C********************************************************************
C      R(ISIZE,JSIZE)=ARRAY TO BE OUTPUT
C      THIS SUBROUTINE IS ABLE TO OUTPUT SELECTED ELEMENTS OF ARRAY
C      IBEG= INITIAL I VALUE TO BE OUTPUT
C      IEND= FINAL I VALUE TO BE OUTPUT
C      IINC= I VALUE INCREMENT FOR OUTPUT
C      (NOTE: (IEND-IBEG)/IINC SHOULD BE AN INTEGER)
C      JBEG,JEND, AND JINC ARE DEFINED SIMILARLY
C      SCALE=SCALE FACTOR
C         IF(SCALE.EQ.0.) SCALE IS CALCULATED TO GIVE BEST DISPLAY
C         ALL ELEMENTS OR ARRAY ARE DIVIDED BY SCALE BEFORE BEING OUTPUT
C      TITLE=CHARACTER STRING THAT IDENTIFIES THE ARRAY BEING OUTPUT
C      NTP= OUTPUT UNIT NUMBER
C      NCOL= NUMBER OF COLUMNS FOR OUTPUT DEVICE (80 OR 132)
C
C*******************************************************************
C
      PARAMETER (JCOL=16)
      DIMENSION R(ISIZE,JSIZE), Y(JCOL)
      REAL*8 SUM0, SUM1, SUM2, TERM, AVE, SD
      CHARACTER TITLE *(*)
C
C  INITIALIZATION
C
      SCALE=XSCALE
      ISCLMX=-12
      SUM0=0.0
      SUM1=0.0
      SUM2=0.0
      MXLINE=57
      JJCOL=(NCOL-4)/8
C
C
C  CALCULATE SIZE OF PRINTOUT ARRAY (NI BY NJ)
C
```

```
      NI=(IEND-IBEG)/IINC+1
      NJ=(JEND-JBEG)/JINC+1
C
C  IF SCALE=0. THEN COMPUTE AUTO SCALE FACTOR
C
      IF(SCALE.EQ.0.) THEN
C        DETERMINE MAXIMUM # OF PLACES TO LEFT OF DECIMAL PT
         DO 10 I=IBEG,IEND,IINC
         DO 10 J=JBEG,JEND,JINC
           IF(R(I,J).NE.0..AND.ABS(R(I,J)).GT.1.E-36) THEN
             TEST=LOG10(ABS(R(I,J)))
             IF(TEST.GE.0.) THEN
                TEST=TEST+.000001
             ELSE
                TEST=TEST-.999999
             END IF

             ITEST=TEST
           ELSE
             ITEST=0
           END IF
           IF(R(I,J).LT.0.) ITEST=ITEST+1
           ISCLMX=MAX0(ISCLMX,ITEST)
   10    CONTINUE
C     DETERMINE SCALE FACTOR SUCH THAT MAX # OF PLACES TO LEFT OF
C     DECIMAL POINT IS 3.
         IPOWER=ISCLMX-2
         SCALE=10.**IPOWER
      ELSE
        IPOWER=NINT(LOG10(SCALE))
      END IF
C
C
C  COMPUTE MEAN AND STD DEVIATION OF OUTPUTTED ARRAY ELEMENTS
C
C
      DO 20 I=1,ISIZE
      DO 20 J=1,JSIZE
        TERM=R(I,J)
        SUM0=SUM0+1.
         IF (ABS(TERM).LT.1.E-20) GO TO 20
        SUM1=SUM1+TERM
        SUM2=SUM2+TERM**2
   20 CONTINUE
      AVE=SUM1/SUM0
      SD=SQRT((SUM2-SUM0*AVE**2)/(SUM0-1.))
C
C
C  OUTPUT DATA
C
C
      JSTART=JBEG
      ISTART=IBEG
      IFINAL=IEND
         IUTT=0
          WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
  810 FORMAT('0',4X,A,'/(1.E',I2,')',4X,'UT=',I6,2X,'AVE=',1PE15.8,2X,
     $       'SD=',E15.8)
      LCOUNT=0
C
C
C  BEGIN OUTPUT LOOP
C
C
```

```
   30 CONTINUE
      JFINAL=(JJCOL-1)*JINC+JSTART
      IF(JFINAL.GT.JEND) JFINAL=JEND
C
C  FOLLOWING IS MAIN IF STATEMENT IN LOOP
C
      IF(JSTART.LE.JEND) THEN
C
C                            OUTPUT A BLOCK
C
          IF(MXLINE-LCOUNT.LE.5) THEN
C
C                                START A NEW PAGE
C
            IUTT=0
            WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
            LCOUNT=0
          END IF
        WRITE(NTP,830) (JJ,JJ=JSTART,JFINAL,JINC)
  830 FORMAT('0',3X,16(5X,I3))
        LCOUNT=LCOUNT+2
        DO 40 II=ISTART,IFINAL,IINC
           LCOUNT=LCOUNT+1
           IF(LCOUNT.GT.MXLINE) THEN
C                                   START A NEW PAGE
               IUTT=0
               WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
               WRITE(NTP,830) (JJ,JJ=JSTART,JFINAL,JINC)
               LCOUNT=2
           END IF
           JCOUNT=0
           DO 50 JJ=JSTART,JFINAL,JINC
               JCOUNT=JCOUNT+1
               Y(JCOUNT)=R(II,JJ)/SCALE
   50      CONTINUE
           WRITE(NTP,840) II,(Y(JJ),JJ=1,JCOUNT)
  840 FORMAT(1X,I3,16(F8.3))
   40   CONTINUE
        JSTART=JFINAL+JINC
        ISTART=IBEG
        GO TO 30
      END IF
      RETURN
      END
      SUBROUTINE HIEPAR
C
C  PURPOSE:  SUBROUTINE TO COMPUTE EMPIRICAL MODEL OF MEV ENERGY
C            ELECTRONS.  THIS ROUTINE IS A DUMMY TO REPRESENT THE
C            TRUE SUBROUTINE TO BE FURNISHED BY D. BAKER OF GSFC.
C
         RETURN
      END
C
      SUBROUTINE OUTPUT
C
C  PURPOSE:  DUMMY SUBROUTINE TO REPRESENT ALGORITHM TO OUTPUT MSM
C            RESULTS TO ENVIRONMENTAL DATA BASE.  THIS ROUTINE WILL BE
C            THE OUTPUT INTERFACE BETWEEN THE MSM AND THE ENVIRONMENTAL
C            DATA BASE.
C
         RETURN
      END
      SUBROUTINE AURL1(FKP,COLAT,ALOCT,LATDIM,LTDIM,AEFLUX,AEMEAN)
C     Determines the number and energy fluxes given Kp, corrected
```

```
C      geomagnetic latitude, and local time                             MSM42880
C                                                                       MSM42890
C     FKP=The most recent Kp value                                      MSM42900
C     COLAT=An array listing pertinent colatitudes (less than 40 deg.)  MSM42910
C      preferably using corrected geomagnetic coordinates               MSM42920
C     ALOCT=An array listing pertinent magnetic local times             MSM42930
C     LATDIM=DIMENSION IN LATITUDE                                      MSM42940
C     LTDIM=DIMENSION IN LOCAL TIME                                     MSM42950
C     AEFLUX=The integral electron energy flux given in units of ergs   MSM42960
C      per cm**2-sec for a given latitude and local time                MSM42970
C     AEMEAN=AVERAGE ENERGY IN EV                                       MSM42980
C     AFLUX=The integral electron number flux given in units of inverse MSM42990
C      cm**2-sec for a given latitude and local time                    MSM43000
C                                                                       MSM43010
      PARAMETER(IIDIM=62,JJDIM=51)                                      MSM43020
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,        MSM43030
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,       MSM43040
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,    MSM43050
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH         MSM43060
C                                                                       MSM43070
      DIMENSION ALOCT(LATDIM,LTDIM),COLAT(LATDIM,LTDIM),                MSM43080
     1 AEFLUX(LATDIM,LTDIM),AFLUX(IIDIM,JJDIM),AEMEAN(LATDIM,LTDIM),    MSM43090
     2 RDC(2),H0DC(2),S1DC(2),S2DC(2),RCOS(6,2),RSIN(6,2),              MSM43100
     3 H0COS(6,2),H0SIN(6,2),S1COS(6,2),S1SIN(6,2),S2COS(6,2),          MSM43110
     4 S2SIN(6,2)                                                       MSM43120
C                                                                       MSM43130
      PI=ACOS(-1.)                                                      MSM43140
C                                                                       MSM43150
C     Read off the model coefficients for the given Kp                  MSM43160
      CALL FNDHDY(FKP,RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,                   MSM43170
     1 S1DC,S1COS,S1SIN,S2DC,S2COS,S2SIN,6,2)                           MSM43180
C                                                                       MSM43190
      DO 20 J=1,LTDIM                                                   MSM43200
        DO 10 I=1,LATDIM                                                MSM43210
C                                                                       MSM43220
          XLT=ALOCT(I,J)                                                MSM43230
C                                                                       MSM43240
C                                                                       MSM43250
C         Determine the values of the energy flux array                 MSM43260
          CALL ADDHDY(RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,S1DC,S1COS,        MSM43270
     1     S1SIN,S2DC,S2COS,S2SIN,6,2,1,XLT,COLAT(I,J),                 MSM43280
     2     AEFLUX(I,J))                                                 MSM43290
C                                                                       MSM43300
C         Convert units from keV/cm**2-ster-sec to erg/cm**2-sec        MSM43310
          AEFLUX(I,J)=(10.**AEFLUX(I,J))*PI*1.6021E-9                   MSM43320
C                                                                       MSM43330
C                                                                       MSM43340
C                                                                       MSM43350
 10     CONTINUE                                                        MSM43360
 20   CONTINUE                                                          MSM43370
C                                                                       MSM43380
      DO 40 J=1,LTDIM                                                   MSM43390
        DO 30 I=1,LATDIM                                                MSM43400
C                                                                       MSM43410
          XLT=ALOCT(I,J)                                                MSM43420
C                                                                       MSM43430
C         Determine the values of the number flux array                 MSM43440
          CALL ADDHDY(RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,S1DC,S1COS,        MSM43450
     1     S1SIN,S2DC,S2COS,S2SIN,6,2,2,XLT,COLAT(I,J),                 MSM43460
     2     AFLUX(I,J))                                                  MSM43470
C                                                                       MSM43480
C         Convert units from (ster-cm**2-sec)**-1 to (cm**2-sec)**-1    MSM43490
          AFLUX(I,J)=(10.**AFLUX(I,J))*PI                               MSM43500
          IF(AFLUX(I,J).NE.0.0)AEMEAN(I,J)=AEFLUX(I,J)/AFLUX(I,J)       MSM43510
```

```
          AEMEAN(I,J)=AEMEAN(I,J)*6.24E11
 30     CONTINUE
 40   CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE FNDHDY(FKP,RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,
     1 S1DC,S1COS,S1SIN,S2DC,S2COS,S2SIN,MDIM,NDIM)
C     Find and read the pertinent Fourier coefficients
C
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1       LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2       LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3       LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION RDC(NDIM),H0DC(NDIM),S1DC(NDIM),S2DC(NDIM),
     1 RCOS(MDIM,NDIM),RSIN(MDIM,NDIM),H0COS(MDIM,NDIM),
     2 H0SIN(MDIM,NDIM),S1COS(MDIM,NDIM),S1SIN(MDIM,NDIM),
     3 S2COS(MDIM,NDIM),S2SIN(MDIM,NDIM)
C
C     DATA IKPCHK /-1/
C
      IFKP2=NINT(FKP)
      IF(IFKP2.GT.6) IFKP2=6
C
C  IF THESE KP COEFS WERE THE LAST READ, DON'T REREAD
C     IF(IKPCHK.EQ.IFKP2) GO TO 99
C
      OPEN(UNIT=LUHDYE,FILE='HARDY',STATUS='OLD')
C
      IF(IREWCK.NE.1)THEN
       GO TO 10
      ELSE IF (IKPCHK.NE.IFKP2)THEN
       REWIND LUHDYE
      ELSE
       GO TO 99
      ENDIF
C
10    IF(IFKP2.EQ.0)GO TO 30
C     Read off irrelevant entries
      DO 20 I=1,IFKP2*26
        READ(LUHDYE,*)COEFF
 20   CONTINUE
C
 30   DO 60 N=1,NDIM
C
C       Read off the constant and trigonometric coefficients
C        for the Epstein-Fourier equation
        READ(LUHDYE,*)RDC(N),H0DC(N),S1DC(N),S2DC(N)
C
        DO 40 M=1,MDIM
          READ(LUHDYE,*)RCOS(M,N),H0COS(M,N),S1COS(M,N),S2COS(M,N)
 40     CONTINUE
C
        DO 50 M=1,MDIM
          READ(LUHDYE,*)RSIN(M,N),H0SIN(M,N),S1SIN(M,N),S2SIN(M,N)
 50     CONTINUE
C
 60   CONTINUE
C
      IKPCHK=IFKP2
      IREWCK=1
```

```
C
      CLOSE(LUHDYE)
C
 99   RETURN
      END
C
C
      SUBROUTINE ADDHDY(RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,S1DC,
     1 S1COS,S1SIN,S2DC,S2COS,S2SIN,MDIM,NDIM,K,ALTIME,COLTDE,FLUX)
C     Calculate the Hardy average value for the given point
C
      DIMENSION RDC(NDIM),H0DC(NDIM),S1DC(NDIM),S2DC(NDIM),
     1 RCOS(MDIM,NDIM),RSIN(MDIM,NDIM),H0COS(MDIM,NDIM),
     2 H0SIN(MDIM,NDIM),S1COS(MDIM,NDIM),S1SIN(MDIM,NDIM),
     3 S2COS(MDIM,NDIM),S2SIN(MDIM,NDIM)
C
      X=90.-ABS(COLTDE)*180./3.14159
      ALCT=12.+ALTIME*12./3.14159
      IF(ALCT.GT.24.) ALCT=ALCT-24.
C
C     Check to see if the Fourier series have to be recalculated
      IF(TEST.EQ.ALCT)GO TO 10
C
C     Determine the coefficients for the Epstein equation
      R=FSUM(RDC,RSIN,RCOS,ALCT,6,2,K)
      H0=FSUM(H0DC,H0SIN,H0COS,ALCT,6,2,K)
      S1=FSUM(S1DC,S1SIN,S1COS,ALCT,6,2,K)
      S2=FSUM(S2DC,S2SIN,S2COS,ALCT,6,2,K)
C
C     Use the Epstein equation to calculate the average value
 10   FLUX=R+S1*(X-H0)+(S2-S1)*ALOG((1-S1/S2*EXP(X-H0))/(1-S1/S2))
C     Set the equatorward lower limit flux value at 6 and
C      the poleward lower limit at 7.
      IF((FLUX.LT.6.).AND.(X.LT.70.))FLUX=6.
      IF((FLUX.LT.7.).AND.(X.GT.70.))FLUX=7.
C
      TEST=ALCT
C
      RETURN
      END
C
C
      REAL FUNCTION FSUM(DC,BSIN,BCOS,ALCT,MDIM,NDIM,K)
C     Determines the sum of a Fourier series
C
      DIMENSION BSIN(MDIM,NDIM),BCOS(MDIM,NDIM),DC(NDIM)
C
      FSUM=0.
C
      DO 10 L=1,MDIM
        FSUM=FSUM+BSIN(L,K)*SIN(L*3.141592*ALCT/12.)+
     1               BCOS(L,K)*COS(L*3.141592*ALCT/12.)
 10   CONTINUE
C
      FSUM=FSUM+DC(K)
C
      RETURN
      END
      SUBROUTINE WRT3D(LUN,IRECMX,ID,RID,CHID,ARRAY,IDIM,JDIM,KDIM,
     1   ITMDIM)
C
C     PROGRAMMER: BOB SPIRO               DATE: 6/26/88
C
C     PURPOSE:  THIS SUBROUTINE WRITES 3-D ARRAYS INTO THE STANDARD
```

```
C     MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
         CHARACTER*80 CHID
         DIMENSION ID(20),RID(20),ARRAY(IDIM,JDIM,KDIM)
C
         IRECL =IDIM*JDIM + 2*80
         OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='UNKNOWN')
C
      ID(12)=KDIM
C
C     WRITE THE ARRAY
C
         DO 10, K=1, KDIM
C
C        CALCULATE THE RECORD NUMBER
C
            ID(13)=K
            IREC = (IRECMX-1)*KDIM + K
            WRITE (LUN,REC=IREC) ID,RID,CHID,
     2                             ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)
 10      CONTINUE
C
         CLOSE (LUN)
C
         RETURN
      END
C
      SUBROUTINE RDHDR(LUN,FILNAM,IRECMX,ID,RID,CHID,IDIM,JDIM,ITMDIM)
C
C     PROGRAMMER: BOB SPIRO                DATE: 6/26/88
C
C     PURPOSE: THIS ROUTINE READS THE HEADER OF THE STANDARD
C     MAGNETOSPERIC SPECIFICATIONS MODEL FILE FORMAT.
C
         CHARACTER CHID*80, FILNAM*6
         DIMENSION ID(20),RID(20)
C
C        OPEN THE FILE
C
         IRECL = IDIM*JDIM + 2*80
         OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
C        READ THE HEADER
C
         READ (LUN,REC=IRECMX) ID,RID,CHID
C
         CLOSE (LUN)
C
         RETURN
      END
C
      SUBROUTINE READ3D(LUN,FILNAM,LREC,IDIM,JDIM,KDIM,ITMDIM,ID,RID,
     2         CHID,ARRAY)
C
C     PROGRAMMER: BOB SPIRO                DATE: 6/26/88
C
C     PURPOSE:  THIS SUBROUTINE READS A LOGICAL RECORD FROM THE
C     STANDARD MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
C
      CHARACTER CHID*80, FILNAM*6
C
         DIMENSION ARRAY(IDIM,JDIM,KDIM)
         DIMENSION ID(20),RID(20)
```

```
C
C        OPEN THE FILE
C
         IRECL = IDIM*JDIM + 2*80
         OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
C        READ THE ARRAY
C
         DO 10, K=1, KDIM
C
C           CALCULATE THE RECORD NUMBER
C
            IREC = (LREC-1)*KDIM + K
            READ (LUN,REC=IREC) ID,RID,CHID,
     2              ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)
 10      CONTINUE
C
         CLOSE(LUN)
C
         RETURN
      END
      SUBROUTINE SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)
C
C THIS SUBROUTINE READS THE NUMBER OF ENERGY CHANNELS AND THEIR
C  ENERGIES AT GEOSYNCHRONOUS ORBIT AND CALCULATES THE ENERGY
C  INVARIANTS NEEDED TO RUN THE PROGRAM.
C THE INPUT ENERGY IS IN EV AND MUST BE IN ASCENDING ORDER SO THAT THE
C  ALMDEL VARIABLES CAN BE CORRECTLY CALCULATED.
C
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LIUDAT,LUHDYE,
     1   LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2   LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3   LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      PARAMETER(VM6=7.0)
C
      DIMENSION ALAM(IEDIM)
      DIMENSION IFLAV(IEDIM)
      DIMENSION ALMDEL(IEDIM)
      DIMENSION NUM(3)
C
      OPEN(UNIT=LUENCH,FILE='ENCHAN',STATUS='OLD')
C
      READ(LUENCH,*)IEMAX
      IF(IEMAX.GT.IEDIM)THEN
       WRITE(6,*)' NUMBER OF ENERGY CHANNELS GREATER THAN MAXIMUM'
       STOP
      ENDIF
C
      DO 5 N=1,3
         NUM(N)=0
    5 CONTINUL
C
      DO 10 IE=1,IEMAX
         READ(LUENCH,*)IFLAV(IE),ENCHNL
         ISP=IFLAV(IE)
         NUM(ISP)=NUM(ISP)+1
         ALAM(IE)= ENCHNL/VM6
         IF(IFLAV(IE).EQ.1)ALAM(IE)=-ALAM(IE)
         IF(NUM(ISP).GT.1.AND.(ABS(ALAM(IE)).LT.ABS(ALAM(IE-1))))THEN
          WRITE(6,*)' ENERGY CHANNELS NOT IN INCREASING ORDER'
          STOP
         ENDIF
   10 CONTINUE
```

```
C
C  COMPUTE ALMDEL
      DO 20 IE=1,IEMAX
        ISP=IFLAV(IE)
        IF(NUM(ISP).GE.2) THEN
          IF(IE.EQ.1.OR.IFLAV(IE-1).NE.ISP) THEN
             ALMMIN=0.
          ELSE
             ALMMIN=(ALAM(IE-1)+ALAM(IE))/2.
          END IF
C
          IF(IE.EQ.IEMAX.OR.IFLAV(IE+1).NE.ISP) THEN
             ALMMAX=1.5*ALAM(IE)-.5*ALAM(IE-1)
          ELSE
             ALMMAX=(ALAM(IE)+ALAM(IE+1))/2.
          END IF
C
          ALMDEL(IE)=ABS(ALMMAX)-ABS(ALMMIN)
C
        END IF
   20 CONTINUE
C
C
      WRITE(6,997)
 997  FORMAT(1H1,'K   ALAM(K)                ALMDEL(K)')
      DO 30 K=1,IEMAX
         WRITE(6,996) K,ALAM(K),ALMDEL(K)
 996  FORMAT(1H ,I3,2F15.4)
 30   CONTINUE
      CLOSE(LUENCH)
      RETURN
      END
      SUBROUTINE BNDSET(LATDIM,LTDIM,IEDIM,ITMDIM,ITMMAX,
     1    IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
     2    NAUGEL,TIMTAG,BNDLOC,IFLAV,AUGPAR,R,P,VM,ETA,ETABND,
     3    FLXBND,ALAM,ALMDEL,IEMAX)
C
C
C  VERSION: 1.00                        DATE: 09.02.89
C                                             05.08.90
C
C  PURPOSE: SUBROUTINE TO COMPUTE BNDY PLASMA DISTRIBUTION BASED ON
C           STATISTICAL (KP-BASED) DATA. THE
C           GENERAL SCHEME IS DESCRIBED IN SECTION 2.5.5 OF
C           CONTRACT #F19628-87-K-0001 FINAL REPORT.
C
C
C  INPUT:
C        LATDIM          NUMBER OF LATITUDINAL GRID PTS
C        LTDIM           NUMBER OF LOCAL TIME GRID POINTS (INCL WRAP)
C        IEDIM           NUMBER OF ENERGY SPECIES
C        ITMDIM          MAX NUMBER OF TEMPORAL GRID PTS
C        ITMMAX          ACTUAL NUMBER
C                        THIS RUN
C        IRDIM           NUMBER OF R BINS IN FLXMAT ARRAY
C        NRGDIM          NUMBER OF LOGARITHMICALLY SPACE ENERGY BINS
C                        IN FLXMAT ARRAY
C        KPDIM           NUMBER OF KP BINS IN FLXMAT ARRAY
C        KPPLUS          AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)
C        ISPDIM          NUMBER OF MASS SPECIES IN FLXMAT ARRAY
C        FLXR            LOG(R) VALUES FOR WHICH FLXMAT IS COMPUTED (RE)
C        FLXNRG          LOG OF ENERGY VALUES FOR WHICH FLXMAT IS
C                        COMPUTED (LOG10(EV))
C        FLXKP           KP VALUES FOR WHICH FLXMAT IS COMPUTED
```

```
C       FLXMAT          ARRAY OF KP-DEPENDENT LOG(FLUX) VALUES          MSM46720
C                       AS FCN OF R AND ENERGY (LOG10(PARTICLES/CM**2/  MSM46730
C                                                     S/SR/EV))         MSM46740
C       NAUGEL          DIMENSION OF AUGMENTED PARRAY, AUGPAR           MSM46750
C       TIMTAG          STRING GIVING TIMES FOR WHICH E AND B FIELD     MSM46760
C                       VALUES ARE CALCULATED                           MSM46770
C       BNDLOC          ARRAY GIVING BNDY LOCATIONS                     MSM46780
C       IFLAV           STRING THAT GIVES WHAT KIND OF PARTICLE         MSM46790
C                       IFLAV=1 FOR ELECTRONS, IFLAV=2 FOR H+,          MSM46800
C                       IFLAV=3 FOR O+ IONS                             MSM46810
C       AUGPAR          AUGMENTED PARAMETER ARRAY                       MSM46820
C       R               RADIAL DISTANCE TO BMIN PT (RE)                 MSM46830
C       P               HOUR ANGLE MEASURED EASTWARD FROM NOON IN X-Y   MSM46840
C                       PLANE OF MAGNETOSPHERE (RADIANS)                MSM46850
C       VM              (FLUX TUBE VOLUME)**-2/3  (RE/NT)**-2/3         MSM46860
C       ETA             INVARIANT DENSITY ARRAY                         MSM46870
C       ALAM            STRING GIVING ENERGY INVARIANTS                 MSM46880
C       ALMDEL          STRING GIVING ENERGY INVARIANT INTERVALS        MSM46890
C       IEMAX           NUMBER OF ENERGY SPECIES FOR RUN                MSM46900
C                                                                       MSM46910
C                                                                       MSM46920
C                                                                       MSM46930
C   OUTPUT:                                                             MSM46940
C       ETABND          BOUNDARY ETA DISTRIBUTION                       MSM46950
C       FLXBND          BOUNDARY FLUX VALUES                            MSM46960
C                                                                       MSM46970
C                                                                       MSM46980
C                                                                       MSM46990
C   PROGRAMMER:  R.W. SPIRO                                             MSM47000
C                                                                       MSM47010
C                                                                       MSM47020
C                                                                       MSM47030
C                                                                       MSM47040
      COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG                              MSM47050
C                                                                       MSM47060
        INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,       MSM47070
     1     EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,         MSM47080
     2     GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,        MSM47090
     3     STAND,PCP,DEQDT,IPATT                                        MSM47100
C                                                                       MSM47110
        COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,     MSM47120
     1     EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,         MSM47130
     2     GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,        MSM47140
     3     STAND,PCP,IPATT,DDST,DEQDT                                   MSM47150
C                                                                       MSM47160
C                                                                       MSM47170
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),ETA(LATDIM,LTDIM,IEDIM)         MSM47180
      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)           MSM47190
      DIMENSION BNDLOC(LTDIM,ITMDIM),TIMTAG(ITMDIM),IFLAV(IEDIM)        MSM47200
      DIMENSION ETABND(LTDIM,IEDIM,ITMDIM),ALAM(IEDIM),ALMDEL(IEDIM)    MSM47210
      DIMENSION FLXBND(LTDIM,IEDIM,ITMDIM)                              MSM47220
      DIMENSION AUGPAR(NAUGEL,ITMDIM)                                   MSM47230
C                                                                       MSM47240
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)                      MSM47250
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)                 MSM47260
C                                                                       MSM47270
      DATA VM13 /2.182/                                                 MSM47280
C                                                                       MSM47290
      PI=ATAN2(0.,-1.)                                                  MSM47300
      WRITE(6,*) 'IN BNDSET'                                            MSM47310
C                                                                       MSM47320
C MAIN DO LOOPS                                                         MSM47330
C                                                                       MSM47340
      ITBEG=1                                                           MSM47350
```

```
      ITEND=ITMMAX
C
      VMBND=VM13
      RBND=13.
      PBND=PI
C
C  LOOP OVER TIME LABELS
      DO 5 IT=ITBEG,ITEND
C
C  LOOP OVER LOCAL TIME COORDINATE
         DO 10 J=2,LTDIM-2
C
            BI=BNDLOC(J,IT)
            BJ=J
            BT=IT
C
C  LOOP OVER ENERGY SPECIES
            DO 20 K=1,IEMAX
               ISP=IFLAV(K)
               IF(IFLAV(K).EQ.2) ATOMWT=1.
               IF(IFLAV(K).EQ.3) ATOMWT=16.
C
C
C
C
               FKP=AUGPAR(KP,IT)
               ENRG=VMBND*ABS(ALAM(K))
C DEBUG 590
C              IF(J.EQ.2) THEN
C                WRITE(6,*)'IT,ISP,FKP,RBND,PBND,ENRG',IT,ISP,FKP,RBND,
C    2                     PBND,ENRG
C              END IF
C
               FLUX=FLXVAL(ISP,FKP,RBND,PBND,ENRG,IRDIM,
     2                    NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
     3                    FLXNRG,FLXKP)
C
C
C
C
C
               FLXBND(J,K,IT)=FLUX
C
C  CONVERT FLUX TO INVARIANT DENSITY
C
               IF(ALAM(K).LT.0.) THEN
                  ETABND(J,K,IT)=FLUX*ALMDEL(K)/7.392E-16/
     2                           SQRT(ABS(ALAM(K)))/VMBND
               ELSE
                  ETABND(J,K,IT)=FLUX*ALMDEL(K)*SQRT(ATOMWT)/1.731E-17/
     2                           SQRT(ABS(ALAM(K)))/VMBND
               END IF
C
C
   20       CONTINUE
C
   10 CONTINUE
C
      DO 15 K=1,IEMAX
         ETABND(1,K,IT)=ETABND(LTDIM-3,K,IT)
         ETABND(LTDIM-1,K,IT)=ETABND(2,K,IT)
         ETABND(LTDIM,K,IT)=ETABND(3,K,IT)
         FLXBND(1,K,IT)=FLXBND(LTDIM-3,K,IT)
         FLXBND(LTDIM-1,K,IT)=FLXBND(2,K,IT)
```

```
          FLXBND(LTDIM,K,IT)=FLXBND(3,K,IT)
   15 CONTINUE
C
    5 CONTINUE
C
C
      RETURN
      END
      FUNCTION CFACT(IDIM,JDIM,IEDIM,IUSE,JUSE,IEUSE,ETAEQK,ALAM,
     2               PWR,ETA,ALAME)
C
C  FUNCTION SUBPROGRAM TO CALCULAT CORRECTION FACTOR BASED ON
C    EQUILIBRIUM ETA VALUE.
C
      CFACT=(1.-ETAEQK/ETA)
      RETURN
      END
      SUBROUTINE SETSCT(L,ISEC,IEDIM,LTDIM,ITMDIM,LATDIM,FKP,ALAM,
     2                  ALOCT,SCTFRC,IEMAX)
C
C  VERSION 2.00                         DATE: 09.11.89
C
C  PURPOSE:
C    SUBROUTINE TO COMPUTE SPECIES-DEPENDENT PITCH
C    ANGLE SCATTERING EFFICIENCY SCTFRC(IE,J,L).
C
C  PROGRAMMER:  R.W. SPIRO
C
C  REF:  SECTION 2.6.3 OF THE FINAL REPORT FOR
C      CONTRACT #F19628-87-K-0001
C
C
      DIMENSION SCTFRC(IEDIM,LTDIM,ITMDIM)
      DIMENSION ALOCT(LATDIM,LTDIM),ALAM(IEDIM)
C
C
      PI=ATAN2(0.,-1.)
C  LOOP OVER ALL LOCAL TIME GRID LINES
      DO 10 J=1,LTDIM
C
C  LOOP OVER ALL ENERGY SPECIES
         DO 20 IE=1,IEMAX
            IF(ALAM(IE).LT.3000.) THEN
C
C              USE KP DEPENDENT FORM, INDEPENDENT OF LOCAL TIME
               IF(FKP.LE.1.5) THEN
                  SCTFRC(IE,J,L)=.333
               ELSE IF(FKP.GT.1.5.AND.FKP.LE.2.5) THEN
                  SCTFRC(IE,J,L)=.333*(2.5-FKP)+.667*(FKP-1.5)
               ELSE
                  SCTFRC(IE,J,L)=.667
               END IF
C
            ELSE
C              USE LOCAL TIME DEPENDENT FORM, INDEPENDENT OF KP
               IF(ALOCT(LATDIM,J).GE.PI/2..AND.ALOCT(LATDIM,J).LE.
     2            3.*PI/2.) THEN
                  SCTFRC(IE,J,L)=0.1/SQRT(ABS(ALAM(IE))/3000.)
               ELSE
                  SCTFRC(IE,J,L)=(0.1+0.9*COS(ALOCT(LATDIM,J)))/
     2                           SQRT(ABS(ALAM(IE))/3000.)
               END IF
C
            END IF
```

```
C
   20    CONTINUE
   10 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE PFIX(P1,P2,P3,P4,P5)
C
C  PURPOSE:  SUBROUTINE TO ADJUST MODULUS OF P2 AND P3 TO MATCH P1.
C          ON OUTPUT P4 CORRESPONDS TO P2, P5 TO P3
C
C  VERSION 1.00                        DATE: 09.04.89
C
C  PROGRAMMER: R.W. SPIRO
C
C
      PI=ATAN2(0.,-1.)
C
      P4=P2
      P5=P3
C
      IF(ABS(P2-P3).GT.PI) THEN
         IF(P2.LT.P3) THEN
            P4=P2+2.*PI
         ELSE
            P5=P3+2.*PI
         END IF
C
         IF(P1.LT.PI) THEN
            P4=P4-2.*PI
            P5=P5-2.*PI
         END IF
C
      END IF
C
C
      RETURN
      END
      SUBROUTINE INITAL(LL,LATDIM,LTDIM,IEDIM,ITMDIM,IRDIM,
     2                  NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,
     3                  FLXMAT,IFLAV,FKP,R,P,VM,ETA,ETABEG,ALAM,ALMDEL,
     4                  BNDLOC,AZERO,BZERO,
     5                  CZERO,IEMAX)
C
C
C  VERSION: 1.00                       DATE: 10.10.89
C
C  PURPOSE: SUBROUTINE TO COMPUTE INITIAL PLASMA DISTRIBUTION BASED ON
C          EITHER STATISTICAL (KP-BASED) DATA OR RESULTS
C          FROM PREVIOUS RUNS.  THE GENERAL SCHEME IS DESCRIBED IN
C          SECTION 2.5.4 OF THE FINAL REPORT FOR
C          CONTRACT #F19628-87-K-0001
C
C  INPUT:
C       LL             TEMPORAL GRID INDEX
C       LATDIM         NUMBER OF LATITUDINAL GRID PTS
C       LTDIM          NUMBER OF LOCAL TIME GRID POINTS (INCL WRAP)
C       IEDIM          MAX NUMBER OF ENERGY SPECIES
C       ITMDIM         MAX NUMBER OF TEMPORAL GRID PTS
C       IRDIM          R DEPTH IN FLXMAT ARRAY
C       NRGDIM         ENERGY DEPTH IN FLXMAT ARRAY
C       KPDIM          KP DEPTH IN FLXMAT ARRAY
C       KPPLUS         AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)
```

```
C        ISPDIM         NUMBER OF MASS SPECIES
C        FLXR           ALOG10(R) VALUES FOR WHICH FLXMAT ARRAY IS
C                        CALCULATED
C        FLXNRG         ALOG10(ENERGY) VALUES FOR WHICH FLXMAT ARRAY IS
C                        CALCULATED
C        FLXKP          KP VALUES FOR WHICH FLXMAT ARRAY IS CALCULATED
C        FLXMAT         ALOG10(FLUX) FOR A GIVEN R,ENERGY, AND KP
C        IFLAV          STRING THAT GIVES WHAT KIND OF PARTICLE
C                       IFLAV=1 FOR ELECTRONS, IFLAV=2 FOR H+,
C                       IFLAV=3 FOR O+ IONS
C        FKP            FLOATING PT KP VALUE
C        R              RADIAL DISTANCE TO BMIN PT (RE)
C        P              HOUR ANGLE MEASURED EASTWARD FROM NOON IN X-Y
C                       PLANE OF MAGNETOSPHERE (RADIANS)
C        VM             (FLUX TUBE VOLUME)**-2/3  (RE/NT)**-2/3
C        ETA            INVARIANT DENSITY ARRAY
C        ALAM           STRING GIVING ENERGY INVARIANTS
C        ALMDEL         STRING GIVING ENERGY INVARIANT INTERVALS
C        BNDLOC         ARRAY GIVING BNDY LOCATIONS
C        AZERO          STATISTICAL MODEL WEIGHTING PARAMETER
C        BZERO          OLD RESULTS WEIGHTING PARAMETER
C        CZERO          TRACEBACK WEIGHTING PARAMETER
C        IEMAX          NUMBER OF PARTICLE SPECIES USED
C
C
C  WEIGHTING PARAMETERS SET IN DATA STATEMENTS
C        GRZERO         PARAMETER TO SPECIFY R GAUSSIAN WIDTH
C        GPZERO         PARAMETER TO SPECIFY P GAUSSIAN WIDTH
C        GLZERO         PARAMETER TO SPECIFY ALAM GAUSSIAN WIDTH
C
C
C  OUTPUT:
C        ETABEG         INITIAL ETA DISTRIBUTION
C
C
C
C  PROGRAMMER:  R.W. SPIRO
C
C
C
C
      COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG
C
C
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),ETA(LATDIM,LTDIM,IEDIM)
      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)
      DIMENSION IFLAV(IEDIM)
      DIMENSION ETABEG(LATDIM,LTDIM,IEDIM),ALAM(IEDIM),ALMDEL(IEDIM)
      DIMENSION BNDLOC(LTDIM,ITMDIM)
C
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)
C
      DATA GRZERO/.05/,GPZERO/.05/,GLZERO/.3/
C
C
      PI=ATAN2(0.,-1.)
C
C  INITIALIZE ETABEG ARRAY
      DO 2 K=1,IEMAX
         DO 4 J=1,LTDIM
            DO 6 I=1,LATDIM
               ETABEG(I,J,K)=0.
    6       CONTINUE
```

```
    4    CONTINUE
    2 CONTINUE
C
C MAIN DO LOOPS
C
      DO 10 K=1,IEMAX
         ISP=IFLAV(K)
         IF(IFLAV(K).EQ.2) ATOMWT=1.
         IF(IFLAV(K).EQ.3) ATOMWT=16.
         DO 20 J=1,LTDIM
            IBEG=BNDLOC(J,LL)
            DO 30 I=IBEG,LATDIM
               SUM1=0.
               SUM2=0.
               ENRG=ABS(ALAM(K))*VM(I,J,LL)
C
C
C
C  PREVIOUS RESULT COMPONENT OF SUM
               IF(ETA(I,J,K).GT.0.) THEN
                  IF(ALAM(K).LT.0.) THEN
                    FLXOLD=7.392E-16*SQRT(ABS(ALAM(K)))*VM(I,J,LL)*
     2                      ETA(I,J,K)/ALMDEL(K)
                  ELSE
                     FLXOLD=1.731E-17*SQRT(ABS(ALAM(K)))*VM(I,J,LL)*
     2                      ETA(I,J,K)/SQRT(ATOMWT)/ALMDEL(K)
                  END IF
C
                  SUM1=SUM1+BZERO
                  SUM2=SUM2+BZERO*FLXOLD
C
               END IF
C
C
C  STATISTICAL (KP-BASED) MODEL COMPONENT OF SUM
C
               SUM1=SUM1+AZERO
               SUM2=SUM2+AZERO*FLXVAL(ISP,FKP,R(I,J,LL),P(I,J,LL),ENRG,
     2              IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
     3              FLXNRG,FLXKP)
C
C
C  CALCULATE DIFFERENTIAL FLUX
C
               IF(SUM1.GT.0.) THEN
                  FLUX=SUM2/SUM1
               ELSE
                  FLUX=0.
               END IF
C
C
C  CONVERT FLUX TO INVARIANT DENSITY
C
               IF(ALAM(K).LT.0.) THEN
                  ETABEG(I,J,K)=FLUX*ALMDEL(K)/7.392E-16/
     2                                 SQRT(ABS(ALAM(K)))/VM(I,J,LL)
               ELSE
                  ETABEG(I,J,K)=FLUX*ALMDEL(K)*SQRT(ATOMWT)/1.731E-17/
     2                                 SQRT(ABS(ALAM(K)))/VM(I,J,LL)
               END IF
C
C
   30       CONTINUE
C
```

```
        DO 32 I=1,IBEG-1
          ETABEG(I,J,K)=ETABEG(IBEG,J,K)
  32    CONTINUE
C
C
  20  CONTINUE
  10 CONTINUE
C
C
    RETURN
    END
    SUBROUTINE TIMINC(ITIME,ITMNEW,IINC)
C
C VERSION 2.00                    DATE: 09.06.89
C
C PURPOSE:  INCREMENT TIME FOR NEXT E AND B FIELD RECORD
C
C
C PROGRAMMER: R.W. SPIRO
C
C
      INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
   1    EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
   2    GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
   3    STAND,PCP,DEQDT,IPATT
C
      COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
   1    EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
   2    GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
   3    STAND,PCP,IPATT,DDST,DEQDT
C
    DIMENSION ITIME(3),ITMNEW(3),IINC(3)
C
C
    ITMNEW(IYEAR)=ITIME(IYEAR)
    ITMNEW(IDAY)=ITIME(IDAY)
C
    ITMNEW(ISECND)=ITIME(ISECND)+IINC(ISECND)
C
    IF(ITMNEW(ISECND).GE.86400) THEN
      ITMNEW(ISECND)=MOD(ITMNEW(ISECND),86400)
      ITMNEW(IDAY)=ITMNEW(IDAY)+1
C
C  -----> CHANGE BEFORE THE YEAR 2100 ******
C
        IF (ITMNEW(IDAY).GT.365.AND.MOD(ITIME(IYEAR),4).NE.0) THEN
          ITMNEW(IDAY) = MOD(ITMNEW(IDAY),365)
          ITMNEW(IYEAR) = ITMNEW(IYEAR) + 1
        ELSEIF(ITMNEW(IDAY).GT.366.AND.MOD(ITIME(IYEAR),4).EQ.0)THEN
          ITMNEW(IDAY) = MOD(ITMNEW(IDAY),366)
          ITMNEW(IYEAR) = ITMNEW(IYEAR) + 1
        ENDIF
C
    END IF
C
C
    RETURN
    END
C
C
C
C
    SUBROUTINE SETREF(LATDIM,LTDIM,IEDIM,ITMDIM,ITMMAX,
   2           IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,TIMTAG,TVAL,IFLAV,
```

```
     3              VM,R,P,BNDLOC,ALAM,ALMDEL,FLXMAT,FLXR,FLXNRG,          MSM51200
     4              FLXKP,FLXREF,IEMAX)                                     MSM51210
C                                                                           MSM51220
C  VERSION 2.00                        DATE: 10.10.89                       MSM51230
C                                            05.08.90                       MSM51240
C                                                                           MSM51250
C  PURPOSE:                                                                 MSM51260
C     SUBROUTINE TO SET UPPER LIMIT REFERENCE FLUX FOR TIME TVAL            MSM51270
C                                                                           MSM51280
C  PROGRAMMER: R.W. SPIRO                                                   MSM51290
C                                                                           MSM51300
C  REF:  SECTION 2.5.6 OF THE FINAL REPORT FOR                              MSM51310
C        CONTRACT #F19628-87-K-0001                                         MSM51320
C                                                                           MSM51330
C      LATDIM      NUMBER OF LATITUDINAL GRID SPACES                        MSM51340
C      LTDIM       NUMBER OF LOCAL TIME (LONGITUDINAL) GRID SPACES          MSM51350
C      IEDIM       MAXIMUM NUMBER OF ENERGY SPECIES                         MSM51360
C      ITMDIM      MAXIMUM NUMBER OF TEMPORAL STEPS                         MSM51370
C      ITMMAX      ACTUAL NUMBER OF TEMPORAL STEPS FOR THIS RUN             MSM51380
C      IRDIM       R DIMENSION OF FLXMAT ARRAYS                             MSM51390
C      NRGDIM      ENERGY DIMENSION OF FLXMAT ARRAYS                        MSM51400
C      KPDIM       KP DIMENSION OF FLXMAT ARRAYS                            MSM51410
C      KPPLUS      KP DIMENSION OF AUGMENTED FLXMAT ARRAYS                  MSM51420
C                  (KPPLUS=KPDIM+2)                                         MSM51430
C      ISPDIM      PLASMA MASS SPECIES DIMENSION                            MSM51440
C      TIMTAG      VECTOR GIVING TIME AT WHICH E AND B FIELD PARAMETERS     MSM51450
C                  ARE COMPUTED (SECONDS)                                   MSM51460
C      TVAL        CURRENT TIME (SECONDS)                                   MSM51470
C      IFLAV       MASS SPECIES FOR A GIVEN PARTICLE ENERGY SPECIES         MSM51480
C                  1=ELECTRONS, 2=H+ IONS, 3=O+ IONS                        MSM51490
C      VM          (FLUX TUBE VOLUME)**(-2/3)                               MSM51500
C      R           ARRAY GIVING RADIAL DISTANCE OF GRID POINTS IN           MSM51510
C                  MAGNETOSPHERIC EQUATORIAL PLANE (RE)                     MSM51520
C      P           LOCAL TIME HOUR ANGLE MEASURED EASTWARD FROM NOON        MSM51530
C                  OF GRID PTS IN MAGNETOSPHERIC EQUATORIAL PLANE           MSM51540
C                  (RADIANS)                                                MSM51550
C      BNDLOC      LOCATION OF OUTER BOUNDARY OF DETAILED MODEL             MSM51560
C      ALAM        ENERGY INVARIANT VECTOR (EV (RE/NT)**2/3)                MSM51570
C      ALMDEL      WIDTH OF ENERGY INVARIANT CHANNELS                       MSM51580
C      FLXMAT      ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,      MSM51590
C                  AND KP                                                   MSM51600
C      FLXR        R VALUES AT WHICH FLXMAT IS CALCULATED                   MSM51610
C      FLXNRG      ENERGY VALUES AT WHICH FLXMAT IS CALCULATED              MSM51620
C      FLXKP       KP VALUES AT WHICH FLXMAT IS CALCULATED                  MSM51630
C      IEMAX       ACTUAL NUMBER OF ENERGY CHANNELS USED                    MSM51640
C                                                                           MSM51650
C                                                                           MSM51660
C  OUTPUT:                                                                  MSM51670
C      FLXREF      UPPER LIMIT REFERENCE FLUX (LOG(#/CM**2 S EV))           MSM51680
C                                                                           MSM51690
      DIMENSION TIMTAG(ITMDIM),ALAM(IEDIM),ALMDEL(IEDIM)                    MSM51700
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),R(LATDIM,LTDIM,ITMDIM)              MSM51710
      DIMENSION P(LATDIM,LTDIM,ITMDIM),FLXREF(LATDIM,LTDIM,IEDIM)           MSM51720
      DIMENSION IFLAV(IEDIM)                                                MSM51730
      DIMENSION BNDLOC(LTDIM,ITMDIM)                                        MSM51740
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)                          MSM51750
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)                     MSM51760
C                                                                           MSM51770
C                                                                           MSM51780
C  CALCULATE NORMALIZED TIME BT                                             MSM51790
      BT=TNORML(TVAL,TIMTAG,ITMDIM)                                         MSM51800
      ITT=BT                                                                MSM51810
      FT=REAL(ITT)                                                          MSM51820
C                                                                           MSM51830
```

```
      C  DO LOOP OVER LOCAL TIME GRID PTS
            DO 10 J=1,LTDIM
               IBEG=BNDLOC(J,ITT)
      C
      C  DO LOOP OVER LATITUDINAL GRID PTS
               DO 20 I=IBEG,LATDIM
                  RVAL=(1.-BT+FT)*R(I,J,ITT)+(BT-FT)*R(I,J,ITT+1)
                  VMVAL=(1.-BT+FT)*VM(I,J,ITT)+(BT-FT)*VM(I,J,ITT+1)
            IF(RVAL.EQ.0.) WRITE(6,*) 'I=',I,' J=',J,' RVAL=',RVAL
            IF(VMVAL.EQ.0.) WRITE(6,*) 'I=',I,' J=',J,' VMVAL=',VMVAL
      C
      C  DO LOOP OVER ENERGY SPECIES
                  DO 30 IE=1,IEMAX
                        ISP=IFLAV(IE)
                        ENRG=ABS(ALAM(IE))*VMVAL
                        FKP=-(KPDIM+2)
                        IF(RVAL.GE.6.6) THEN
                           FLXREF(I,J,IE)=FLXVAL(ISP,FKP,RVAL,PP,ENRG,IRDIM,
           2               NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,FLXNRG,
           3               FLXKP)
                        ELSE
                           EGEO=ENRG*(RVAL/6.6)**(8./3.)
                           FLXGEO=FLXVAL(ISP,FKP,6.6,PP,EGEO,IRDIM,
           2                 NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,FLXNRG,
           3                 FLXKP)
      C
                           FLXREF(I,J,IE)=((6.6/RVAL)**(8./3.))*FLXGEO
      C
                        END IF
      C
      C
         30       CONTINUE
         20    CONTINUE
         10 CONTINUE
      C
      C
            RETURN
            END
            SUBROUTINE EFLOC(RVAL,XLT,LL,LATDIM,LTDIM,ITMDIM,
           2                  R,ALOCT,BI,BJ)
      C
      C  PURPOSE:  FIND GRID LOCATION (BI,BJ) OF PHYSICAL LOCATIONS GIVEN BY
      C            (RVAL,XLT,LL)
      C
      C  VERSION 1.00                        DATE: 09.13.89
      C
      C
      C  INPUT:
      C        RVAL      RADIAL DISTANCE (RE)
      C        XLT       HOUR ANGLE MEASURED EASTWARD FROM NOON (RADIAN)
      C        LL        TIME INDEX
      C        LATDIM    NUMBER OF LATITUDINAL GRID PTS
      C        LTDIM     MAX NUMBER OF LOCAL TIME GRID PTS (INCL WRAPAROUND)
      C        ITMDIM    MAX NUMBER OF TEMPORAL GRID PTS
      C                  PARAMETERS ARE CALCULATED
      C        R         RADIAL DISTANCE ARRAY (RE)
      C        ALOCT     HOUR ANGLE ARRAY (RADIANS)
      C
      C  OUTPUT:
      C        BI        NON-INTEGER I LOCATION OF (RVAL,XLT)
      C        BJ        NON-INTEGER J LOCATION OF (RVAL,XLT)
      C
      C  PROGRAMMER: R.W. SPIRO
      C
```

```
C
C
C
C
C
      DIMENSION R(LATDIM,LTDIM,ITMDIM),ALOCT(LATDIM,LTDIM)
C
C
      PI=ATAN2(0.,-1.)
C
C
C
C
C
C
C
C SEARCH GRID AND TRY TO LOCALIZE (RVAL,XLT) WITHIN A GRID SQUARE
      DO 30 J=2,LTDIM-2
         JJ=J
C
         DO 40 I=2,LATDIM
            II=I
C
C GET XLT,ALOCT(I,J), AND ALOCT(I,J+1) IN SAME MODULUS
            CALL PFIX(XLT,ALOCT(I,J),ALOCT(I,J+1),P4,P5)
C           WRITE(6,*) I,J,RVAL,R(I-1,J,LL),R(I,J,LL),XLT,P4,P5
C
            IF(RVAL.LE.R(I-1,J,LL)
     2         .AND.
     3             RVAL.GT.R(I,J,LL)
     4         .AND.
     5             XLT.GE.P4
     6         .AND.
     7             XLT.LT.P5)
     8      THEN
               GO TO 50
            END IF
C
   40    CONTINUE
   30 CONTINUE
C
      WRITE(6,*) 'STOPPING IN EFLOC'
      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
      WRITE(6,*) 'RVAL= ',RVAL,'  XLT= ',XLT
      STOP
C
C
   50 CONTINUE
C
      DO 45 IK=2,LATDIM
         IIK=IK
         IF(RVAL.LE.R(IK-1,J+1,LL).AND.RVAL.GT.R(IK,J+1,LL)) GO TO 60
   45 CONTINUE
C
      WRITE(6,*) 'STOPPING IN EFLOC'
      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
      WRITE(6,*) 'RVAL= ',RVAL,'  XLT= ',XLT
      STOP
C
   60 CONTINUE
C
C
C COMPUTE COEFFICIENTS FOR INTERPOLATION
      F1=(RVAL-R(II-1,JJ,LL))/(R(II,JJ,LL)-R(II-1,JJ,LL))+FLOAT(II-1)
```

```
      F2=(RVAL-R(IIK-1,JJ+1,LL))/(R(IIK,JJ+1,LL)-R(IIK-1,JJ+1,LL))+
     2    FLOAT(IIK-1)
C
      CALL PFIX(XLT,ALOCT(II,JJ),ALOCT(IIK,JJ+1),P4,P5)
C
      BJ=(XLT-P4)/(P5-P4)+FLOAT(JJ)
C
      BI=F1+(BJ-FLOAT(JJ))*(F2-F1)/(FLOAT(JJ+1)-FLOAT(JJ))
C
C
      RETURN
      END
C
C
C
      SUBROUTINE EFBNDY(LATDIM,LTDIM,ITMDIM,LL,EQEDG,STDOFF,VDROP,R,
     2                  ALOCT,COLAT,A,B,DX,DY)
C
C  VERSION 1.00                     DATE: 09.13.89
C          1.01                           10.25.89
C  PURPOSE:
C   SUBROUTINE TO CALCULATE LATITUDES AND WIDTHS OF EFIELD BNDYS 2 & 3
C
C  PROGRAMMER: R.W. SPIRO
C
C  REF: SECTION 2.4.2 OF THE FINAL REPORT FOR
C       CONTRACT #F19628-87-K-0001
C
C
      DIMENSION R(LATDIM,LTDIM,ITMDIM),ALOCT(LATDIM,LTDIM)
      DIMENSION COLAT(LATDIM,LTDIM)
      DIMENSION A(3),B(3),DX(3),DY(3)
      DIMENSION XLAMDA(4,3),DLAMDA(4)
C
C
      PI=ATAN2(0.,-1.)
C
      WRITE(6,*) 'IN EFBNDY EQEDG= ',EQEDG,'STDOFF= ',STDOFF
C
      RNOON=.95*STDOFF
      RDAWN=1.4*STDOFF
      RDUSK=1.4*STDOFF
      RMIDNT=2.0*STDOFF
C
C
C  XLAMDA(1,N) REFERS TO NOON; XLAMDA(2,N) REFERS TO DUSK
C  XLAMDA(3,N) REFERS TO MIDNIGHT; XLAMDA(4,N) REFERS TO DAWN
C  WHERE N=1,2,3 REFERS TO THE THREE BNDYS OF THE E FIELD CALCULATION
C  BNDY 1 IS CALCULATED INTERNALLY IN EFIELD
C  BNDYS 2 AND 3 ARE CALCULATED HERE
C
      XLAMDA(3,3)=EQEDG
      XLAMDA(2,3)=EQEDG+1.
      XLAMDA(4,3)=EQEDG+1.
      XLAMDA(1,3)=EQEDG+((66.95-EQEDG)*7.725+2.8*(EQEDG-56.8))/10.15
C
C
C  CALCULATION OF PARAMETERS USED IN CALCULATING BNDY 2
C
C  LOCATE RNOON,RDAWN, RDUSK, AND RMIDNT IN OUR GRID, IE., FIND
C    (BINOON,BJNOON), (BIDUSK,BJDUSK), (BIMDNT,BJMDNT), (BIDAWN,BJDAWN)
C
      TNOON=0.
      CALL EFLOC(RNOON,TNOON,LL,LATDIM,LTDIM,ITMDIM,R,ALOCT,
```

```
     2             BINOON,BJNOON)
C
      TDUSK=PI/2.
      CALL EFLOC(RDUSK,TDUSK,LL,LATDIM,LTDIM,ITMDIM,R,ALOCT,
     2             BIDUSK,BJDUSK)
C
C
      TMIDNT=PI
      CALL EFLOC(RMIDNT,TMIDNT,LL,LATDIM,LTDIM,ITMDIM,R,ALOCT,
     2             BIMDNT,BJMDNT)
C
      TDAWN=3.*PI/2.
      CALL EFLOC(RDAWN,TDAWN,LL,LATDIM,LTDIM,ITMDIM,R,ALOCT,
     2             BIDAWN,BJDAWN)
C
C
      XLAT1=G3NTRP(COLAT,LATDIM,LTDIM,1,BINOON,BJNOON,1.)
      XLAMDA(1,2)=(PI/2.-XLAT1)*180./PI
C
      XLAT2=G3NTRP(COLAT,LATDIM,LTDIM,1,BIDUSK,BJDUSK,1.)
      XLAMDA(2,2)=(PI/2.-XLAT2)*180./PI
C
      XLAT3=G3NTRP(COLAT,LATDIM,LTDIM,1,BIMDNT,BJMDNT,1.)
      XLAMDA(3,2)=(PI/2.-XLAT3)*180./PI
C
      XLAT4=G3NTRP(COLAT,LATDIM,LTDIM,1,BIDAWN,BJDAWN,1.)
      XLAMDA(4,2)=(PI/2.-XLAT4)*180./PI
C
C  CHECK BOUNDARY 3 (EQ. EDGE OF SUNWARD FLOW) FOR RIDICULOUS VALUE;
C    CHANGE IF NECESSARY
C
      EMAX=.1
      DLAMDA(1)=3.
      DLAMDA(3)=3.
      DLAMDA(2)=4.5E-6*VDROP/EMAX
      DLAMDA(4)=DLAMDA(2)
C
C  LOOP OVER NOON, DUSK, MIDNIGHT, AND DAWN
      DO 8 II=1,4
         DELTA=XLAMDA(II,2)-DLAMDA(II)-XLAMDA(II,3)
         IF(DELTA.LT.0.) THEN
            WRITE(6,*) 'ADJUSTING XLAMDA(II,3) IN EFBNDY'
            WRITE(6,*) 'LL=',LL,' II=',II,' XLAMDA(OLD)=',XLAMDA(II,3)
            XLAMDA(II,3)=XLAMDA(II,3)+DELTA
            WRITE(6,*) '                 XLAMDA(NEW)=',XLAMDA(II,3)
         END IF
    8 CONTINUE
C
C
      DO 10 II=2,3
         DX(II)=-.5*(XLAMDA(1,II)-XLAMDA(3,II))
         DY(II)=-.5*(XLAMDA(2,II)-XLAMDA(4,II))
C
         X=.5*(180.-XLAMDA(1,II)-XLAMDA(3,II))
         Y=.5*(180.-XLAMDA(4,II)-XLAMDA(2,II))
C
         A(II)=SQRT(((X*Y)**2-(DX(II)*DY(II))**2)/(Y**2-DY(II)**2))
         B(II)=SQRT(((X*Y)**2-(DX(II)*DY(II))**2)/(X**2-DX(II)**2))
C
   10 CONTINUE
C
C
      RETURN
      END
```

```
C
C ************************** F E C O N **************************
C
      SUBROUTINE FECON(ISTART,IINC,IEND,ITMDIM,NAUGEL,MODE,AUGPAR,
     1                 ITMMAX,TIMTAG)
C
C  VERSION 1.0                  DATE: JANUARY 9, 1988
C          2.0                        SEPTEMBER 14, 1989
C
C  PROGRAMMER: BRYAN A. BALES
C
C  PURPOSE:  FECON CALLS PARGEN TO OBTAIN TIME-NORMALIZED OBSERVED DATA
C         THAT IS THEN USED TO CALCULATE OTHER VALUES NEEDED BY THE
C         EFIELD AND BFIELD ROUTINES.
C
      PARAMETER (NELTS=28,ITDIM=50)
C
C
      INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,DEQDT,IPATT
C
      LOGICAL*1 MODE(NAUGEL,ITMDIM)
      COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,IPATT,DDST,DEQDT
C
C
      DIMENSION PARRAY(NELTS,ITDIM),AUGPAR(NAUGEL,ITMDIM),ITIME(3)
      DIMENSION ITMNEW(3),ISTART(3),IINC(3),IEND(3)
      DIMENSION TIMTAG(ITMDIM)
C
      IF (ITDIM.NE.ITMDIM) STOP 'STOPPING IN FECON. ITDIM.NE.ITMDIM'
C
      DO 30 IDO=1,NAUGEL
       DO 40 JDO=1,ITMDIM
        MODE(IDO,JDO)=.FALSE.
40     CONTINUE
30    CONTINUE
C
      CALL PARGEN(ISTART,IINC,IEND,ITMDIM,NELTS,MODE,PARRAY,
     1              ITMMAX,TIMTAG,NAUGEL)

      ITIME(1) = ISTART(1)
      ITIME(2) = ISTART(2)
      ITIME(3) = ISTART(3)
C
      DO 20, ITM=1, ITMMAX
         DO 10, I=1, NELTS
            AUGPAR(I,ITM) = PARRAY(I,ITM)
 10      CONTINUE
C
         CALL TILT(ITIME(IDAY),ITIME(IYEAR),AUGPAR(TILTW,ITM))
C
         CALL STNDFL(MODE(STAND,ITM),MODE(SWVEL,ITM),
     1      MODE(SWDEN,ITM),PARRAY(SWVEL,ITM),PARRAY(SWDEN,ITM),
     2      PARRAY(KP,ITM),AUGPAR(STAND,ITM))
C
         CALL DSTDFL(MODE(DST,ITM),AUGPAR,NAUGEL,ITMDIM,ITM,ITMMAX,
     1      TIMTAG)
C
         CALL EQTDFL(MODE(EQEDGE,ITM),AUGPAR,NAUGEL,ITMDIM,ITM,
```

```
     1                    ITMMAX,TIMTAG)
C
            CALL PCPDFL(MODE(PCP,ITM),PARRAY(KP,ITM),AUGPAR(PCP,ITM))
C
            CALL PATDFL(MODE(IPATT,ITM),MODE(IMFBY,ITM),
     1                  MODE(IMFBZ,ITM),PARRAY(IMFBY,ITM),
     2                  PARRAY(IMFBZ,ITM),AUGPAR(IPATT,ITM))
C
            CALL CLPDFL(MODE(CLAPSE,ITM),AUGPAR(CLAPSE,ITM))
C
C
            CALL TIMINC(ITIME,ITMNEW,IINC)
            ITIME(1) = ITMNEW(1)
            ITIME(2) = ITMNEW(2)
            ITIME(3) = ITMNEW(3)
 20      CONTINUE
C
         RETURN
      END
C
C **************************** P A R G E N ***********************
C
      SUBROUTINE PARGEN(ISTART,IINC,IEND,ITMDIM,NELTS,MODE,PARRAY,
     1                  ITMMAX,TIMTAG,NAUGEL)
C
C  VERSION 1.0                    DATE: JANUARY 9, 1988
C          2.0                          SEPTEMBER 14, 1989
C
C  PROGRAMMER: BRYAN A. BALES
C
C  PURPOSE:  PARGEN GETS OBSERVED VALUES FROM THE ENVIRONMENTAL DATABASE
C         AND PERFORMS AN INTERPOLATION OR EXTRAPOLATION AS APPROPRIATE
C         TO PROVIDE DATA AT A NORMALIZED TIME.
C
         PARAMETER (NPELTS=28,NDIM=80,NAUGL=28)
C
         CHARACTER*6 PARAM(NPELTS)
C
         LOGICAL*1 MODE(NAUGEL,ITMDIM)
C
         INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,DEQDT,IPATT
C
         DATA PARAM /' ',' ',' ',' ',' ','FKP','DST','EQEDGE',' ',' ',
     1        ' ','IMFBX','IMFBY','IMFBZ','CLAPSE',' ',' ',' ',' ',' ',
     2        'SWVEL','SWDEN',' ',' ','PCP','XIPATT','DDST','DEQDT'/
C
C
         COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3      STAND,PCP,IPATT,DDST,DEQDT
C
         DIMENSION PARRAY(NELTS,ITMDIM),ISTART(3),IINC(3),IEND(3)
         DIMENSION DARRY(7,NDIM),ITIME(3),ITMNEW(3),IGAP(NAUGL)
         DIMENSION TIMTAG(ITMDIM),XA(NDIM),YA(NDIM),XKPMAT(3,NDIM)
C
         DATA IGAP /5*0,86400,86400,28800,16*0,39744,31104,2*0/
C
         IF (NPELTS.NE.NELTS) STOP 'STOPPING IN PARGEN. NELTS.NE.NPELTS'
         IF(NAUGL.NE.NAUGEL) STOP 'STOPPING IN PARGEN.  NAUGL.NE.NAUGEL'
C
```

```
      REND = TCONV3(IEND,ISTART)
      NODATA = 0
C
      DO 10, I=6, NELTS
         ITM = 1
         ITIME(1) = ISTART(1)
         ITIME(2) = ISTART(2)
         ITIME(3) = ISTART(3)
C
      IF (I.EQ.KP.OR.I.EQ.DST.OR.I.EQ.PCP.OR.I.EQ.IPATT.OR.
     1     I.EQ.CLAPSE.OR.I.EQ.SWDEN.OR.I.EQ.SWVEL.OR.
     2     I.EQ.EQEDGE) THEN
C
         CALL INDATA(PARAM(I),ISTART,IEND,NDIM,DARRY,NUMNUM)
         WRITE(6,*) 'COMING OUT OF INDATA, I= ',I,' NUMNUM= ',NUMNUM
         IF((NUMNUM.EQ.0).AND.(I.EQ.KP))THEN
          WRITE(6,*)' NO KP AVAILABLE TO RUN PROGRAM.'
          STOP
         ENDIF
C
         STARTY = REAL(ISTART(1))
         IF(ISTART(3).LT.86400) THEN
            STARTD = REAL(ISTART(2)) + REAL(ISTART(3))/86400.
         ELSE IF(ISTART(3).EQ.86400) THEN
            STARTD=REAL(ISTART(2))
         ELSE
            WRITE(6,*) 'INCORRECT START TIME HAS BEEN INPUT'
            WRITE(6,*) 'STOPPING PROGRAM IN PARGEN'
            STOP
         END IF
C
         CALL SMOOTH(DARRY,NDIM,NUMNUM,STARTY,STARTD,NA,XA,YA,IERR)
C
         WRITE(6,*) ' NA= ',NA,' NUMNUM=',NUMNUM,' NDIM=',NDIM
         DO 42 NX=1,NA
            WRITE(6,*) ' TIME=',XA(NX),'    VALUE=',YA(NX)
   42    CONTINUE
C
         WRITE (6,*) 'RETURNED FROM SMOOTH FOR PARAMETER ',PARAM(I)
C
         IF (IERR.NE.0) THEN
            WRITE (6,*) 'TIME REVERSAL IN PARAMETER ',PARAM(I)
            STOP 'SERIOUS ERROR IN ENVIRONMENTAL DATA.'
         ENDIF
         IF(NUMNUM.EQ.0)GO TO 10
C
  20     CONTINUE
            IF (I.EQ.KP) THEN
               PARRAY(IYEAR,ITM) = ITIME(IYEAR)
               PARRAY(IDAY,ITM) = ITIME(IDAY)
               PARRAY(IPHOUR,ITM) = REAL(ITIME(ISECND)/3600)
               PARRAY(IPMIN,ITM) = REAL(MOD(ITIME(ISECND),3600)/60)
               PARRAY(IPSEC,ITM) = REAL(MOD(ITIME(ISECND),60))
               TIMTAG(ITM) = TCONV3(ITIME,ISTART)
               WRITE(6,*) 'TIMTAG(',ITM,') = ',TIMTAG(ITM)
            ENDIF
C
            IF (I.EQ.IPATT) THEN
               CALL DTXIPT(TIMTAG(ITM),NA,XA,YA,Y,DYDX,DELTA,IERR)
               PARRAY(I,ITM) = Y
            ELSEIF (I.NE.DDST.AND.I.NE.DEQDT) THEN
               CALL DTNTRP(TIMTAG(ITM),NA,XA,YA,Y,DYDX,DELTA,IERR)
C
               PARRAY(I,ITM) = Y
```

```
                  IF((DELTA.LE.IGAP(I)).OR.(IGAP(I).EQ.0))THEN
                     MODE(I,ITM) = .TRUE.
                  ELSE
                     MODE(I,ITM) = .FALSE.
                  ENDIF
C
C  DYDX IS IN UNITS OF Y/SEC
C  NEED TO CHANGE TO UNITS OF Y/HOUR
                  IF (I.EQ.DST) PARRAY(DDST,ITM) = DYDX*3600.
                  IF (I.EQ.EQEDGE) PARRAY(DEQDT,ITM) = DYDX*3600.
                   IF(I.EQ.DST)
     1              WRITE(6,*)' ITM,DST,DYDX ',TIMTAG(ITM),
     1               PARRAY(DST,ITM),PARRAY(DDST,ITM)
                   IF(I.EQ.EQEDGE)
     2              WRITE(6,*)' ITM,EQED,DYDX',TIMTAG(ITM),
     1               PARRAY(EQEDGE,ITM),PARRAY(DEQDT,ITM)
                ENDIF
C
                ITM = ITM + 1
                CALL TIMINC(ITIME,ITMNEW,IINC)
                ITIME(1) = ITMNEW(1)
                ITIME(2) = ITMNEW(2)
                ITIME(3) = ITMNEW(3)
C
              IF (NINT(TCONV3(ITIME,ISTART)).LE.NINT(REND)) GO TO 20
C
              IF (I.EQ.KP) ITMMAX = ITM-1
            ENDIF
C
 10         CONTINUE
C
            RETURN
          END
C
C   ****************************** INDATA ******************************
C
          SUBROUTINE INDATA(PARAM,STARTT,ENDT,NDIM,DARRY,NUMNUM)
C
C   THIS IS A DUMMY SUBROUTINE TO READ IN DATA FOR THE
C   MAGNETOSHPERIC SPECIFICATION MODEL DURING DEVELOPMENT
C   AND TESTING.  THIS ROUTINE WILL BE REPLACED IN THE
C   OPERATIONAL MSM WITH AN INTERFACE ROUTINE TO THE
C   AWS ENVIRONMENTAL DATABASE.
C
C   BRYAN BALES               9/12/89
C
          CHARACTER*(*) PARAM
C
          DIMENSION STARTT(3),ENDT(3),DARRY(7,NDIM)
C
            NUMNUM = 1
            OPEN (UNIT=98,FILE=PARAM,STATUS='OLD')
C
 10         CONTINUE
              READ (98,*,END=20)        (DARRY(I,NUMNUM),I=1,7)
              NUMNUM = NUMNUM + 1
            IF (NUMNUM.LE.NDIM) GO TO 10
C
            STOP 'Stopping in INDATA. NUMNUM is greater than NDIM.'
C
 20         CONTINUE
            NUMNUM = NUMNUM - 1
            CLOSE (98)
C
```

```
         RETURN
      END
C
C   ****************************** GETMAT ******************************
C
      SUBROUTINE GETMAT(IWANT,LATDIM,LTDIM,BFPAR,WORK,BFLIM,BFEXST)
C
C   THIS SUBROUTINE FINDS THE MAGNETIC FIELD MATRICES NEEDED TO
C   INTERPOLATE ONE MATRIX THAT IS REPRESENTATIVE OF CURRENT
C   CONDITIONS.
C
C   BRYAN BALES            9/12/89
C
      PARAMETER (IMSTND=5,IMTILT=5,IMINED=16,IMDST=8,IMSTCH=2)
C
      INTEGER BFNDX(2,5)
      LOGICAL*1 BFEXST(2,2,2,2,2)
      LOGICAL MEXIST
C
      DIMENSION STNDPR(IMSTND),TILTPR(IMTILT),FINEDP(IMINED)
      DIMENSION DSTPR(IMDST),STCHPR(IMSTCH)
      DIMENSION BFLIM(2,5),BFPAR(5)
      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
      DIMENSION XDEB(62,51),YDEB(62,51),ZDEB(62,51)
      DATA MSTND/IMSTND/
      DATA MTILT/IMTILT/
      DATA MINED/IMINED/
      DATA MDST/IMDST/
      DATA MSTCH/IMSTCH/
C
      DATA STNDPR/6.0,8.0,10.0,12.0,14.0/
      DATA TILTPR/-35.0,-17.5,0.0,17.5,35.0/
      DATA FINEDP/49.47, 53.20, 55.89, 57.96, 59.60,
     1           60.95, 62.06, 63.02, 63.99, 65.01,
     2           65.91, 66.72, 67.45, 68.12, 68.73,
     3           69.30/
      DATA DSTPR/-400.0, -300.0, -200.0, -150.0,
     1         -100.0,  -50.0,    0.0,   50.0/
      DATA STCHPR/0.0, 1.0/
C
      EXTERNAL MEXIST
C
      CALL FNDBRK(BFPAR(1),STNDPR,MSTND,BFNDX(1,1),BFNDX(2,1))
      CALL FNDBRK(BFPAR(2),TILTPR,MTILT,BFNDX(1,2),BFNDX(2,2))
      CALL FNDBRK(BFPAR(3),FINEDP,MINED,BFNDX(1,3),BFNDX(2,3))
      CALL FNDBRK(BFPAR(4),DSTPR, MDST, BFNDX(1,4),BFNDX(2,4))
      CALL FNDBRK(BFPAR(5),STCHPR,MSTCH,BFNDX(1,5),BFNDX(2,5))
C
      DO 5, I=1,2
         BFLIM(I,1) = STNDPR(BFNDX(I,1))
         BFLIM(I,2) = TILTPR(BFNDX(I,2))
         BFLIM(I,3) = FINEDP(BFNDX(I,3))
         BFLIM(I,4) = DSTPR(BFNDX(I,4))
         BFLIM(I,5) = STCHPR(BFNDX(I,5))
 5    CONTINUE
C
C
      DO 10, I=1,2
         DO 20, J=1,2
            DO 30, K=1,2
               DO 40, L=1,2
                  DO 50, M=1,2
                     IF (MEXIST(BFNDX(I,1),BFNDX(J,2),BFNDX(K,3),
     1                          BFNDX(L,4),BFNDX(M,5))) THEN
```

```
                  CALL LOADBM(IWANT,LATDIM,LTDIM,I,J,K,L,M,BFNDX,  MSM57600
     1                          WORK)                              MSM57610
                  BFEXST(I,J,K,L,M) = .TRUE.                       MSM57620
               ELSE                                                MSM57630
                  CALL ZEROBM(LATDIM,LTDIM,I,J,K,L,M,WORK)         MSM57640
                  BFEXST(I,J,K,L,M) = .FALSE.                      MSM57650
               ENDIF                                               MSM57660
C                                                                  MSM57670
               IF(BFNDX(J,2).EQ.3.AND.IWANT.EQ.1) THEN             MSM57680
                 IF((3-BFNDX(M,5)).EQ.1.OR.BFPAR(5).LT.1) THEN     MSM57690
C                                                                  MSM57700
C                WRITE (6,*) 'STANDOFF =',BFLIM(I,1),' TILT =',    MSM57710
C    1                     BFLIM(J,2),' EQEDGE =',BFLIM(K,3),      MSM57720
C    2                     ' DST =',BFLIM(L,4),' COLLAPSE',        MSM57730
C    3                     BFLIM(M,5)                              MSM57740
C                WRITE (6,1001) BFNDX(J,2),BFNDX(I,1),BFNDX(L,4),  MSM57750
C    1                        BFNDX(K,3),(3-BFNDX(M,5))            MSM57760
C1001            FORMAT (' BF',I1,I1,I1,I2.2,I1)                   MSM57770
                 WRITE(6,1001) BFNDX(J,2),BFNDX(I,1),BFNDX(L,4),   MSM57780
     2                BFNDX(K,3),(3-BFNDX(M,5)),BFEXST(I,J,K,L,M), MSM57790
     3                BFLIM(I,1),BFLIM(J,2),BFLIM(K,3),            MSM57800
     4                BFLIM(L,4),BFLIM(M,5)                        MSM57810
 1001 FORMAT(' BF',I1,I1,I1,I2.2,I1,1X,L1,3X,'STANDOFF= ',F8.2,3X, MSM57820
     2       'TILT= ',F8.2,3X,'EQEDGE= ',F8.2,3X,'DST= ',F8.2,3X,  MSM57830
     3       'COLLAPSE= ',F4.2)                                    MSM57840
                 END IF                                            MSM57850
               END IF                                              MSM57860
C              IF((IWANT.EQ.1).AND.(BFEXST(I,J,K,L,M)))THEN        MSM57870
C               DO 999 LAT=1,LATDIM                                MSM57880
C                DO 998 LT=1,LTDIM                                 MSM57890
C                 XDEB(LAT,LT)=WORK(LAT,LT,I,J,K,L,M)              MSM57900
998             CONTINUE                                           MSM57910
999            CONTINUE                                            MSM57920
C              CALL OUTP(XDEB,LATDIM,LTDIM,10,LATDIM,1,3,          MSM57930
C    1            LTDIM-3,1,1.,'XDEB',6,132)                       MSM57940
C             ENDIF                                                MSM57950
C             IF((IWANT.EQ.2).AND.(BFEXST(I,J,K,L,M)))THEN         MSM57960
C             DO 997 LAT=1,LATDIM                                  MSM57970
C              DO 996 LT=1,LTDIM                                   MSM57980
C               YDEB(LAT,LT)=WORK(LAT,LT,I,J,K,L,M)                MSM57990
996            CONTINUE                                            MSM58000
997           CONTINUE                                             MSM58010
C             CALL OUTP(YDEB,LATDIM,LTDIM,10,LATDIM,1,3,           MSM58020
C    1            LTDIM-3,1,1.,'YDEB',6,132)                       MSM58030
C             ENDIF                                                MSM58040
C             IF((IWANT.EQ.3).AND.(BFEXST(I,J,K,L,M)))THEN         MSM58050
C              DO 995 LAT=1,LATDIM                                 MSM58060
C               DO 994 LT=1,LTDIM                                  MSM58070
C                ZDEB(LAT,LT)=WORK(LAT,LT,I,J,K,L,M)               MSM58080
994             CONTINUE                                           MSM58090
995            CONTINUE                                            MSM58100
C              CALL OUTP(ZDEB,LATDIM,LTDIM,10,LATDIM,1,3,          MSM58110
C    1             LTDIM-1,1,1.,'ZDEB',6,132)                      MSM58120
C            ENDIF                                                 MSM58130
C                                                                  MSM58140
 50            CONTINUE                                            MSM58150
 40          CONTINUE                                              MSM58160
 30        CONTINUE                                                MSM58170
 20      CONTINUE                                                  MSM58180
 10    CONTINUE                                                    MSM58190
C                                                                  MSM58200
       RETURN                                                      MSM58210
       END                                                         MSM58220
C                                                                  MSM58230
```

```
C    ****************************** FNDBRK ******************************
C
      SUBROUTINE FNDBRK(PARVAL,PVALS,IPDIM,MIN,MAX)
C
C   THIS SUBROUTINE FINDS THE INDICES OF THE VALUES IN ARRAY
C   PVALS THAT BRACKET PARVAL.
C
C   BRYAN BALES                9/12/89
C
      DIMENSION PVALS(IPDIM)
C
         MIN = 0
C
 10      CONTINUE
            MIN = MIN + 1
            MAX = MIN + 1
C        IF (PARVAL.LE.PVALS(MIN)) THEN
C           MAX = MIN
C        ELSEIF (PARVAL.GT.PVALS(MAX).AND.MAX.EQ.IPDIM) THEN
C           MIN = MAX
         IF (PARVAL.GT.PVALS(MAX).AND.MAX.LT.IPDIM) THEN
            GO TO 10
         ENDIF
C
         RETURN
      END
C
C    **************************** MEXIST ***************************
C
      LOGICAL FUNCTION MEXIST(MSTND,MTILT,MINED,MDST,MSTCH)
C
C   THIS FUNCTION CHECKS TO SEE IF A PARTICULAR B-MATRIX EXISTS
C   IN THE B-SUPERMATRIX(TM).
C
         CHARACTER*8 FILNAM
         LOGICAL TEST
C
         WRITE (FILNAM,1000) MTILT,MSTND,MDST,MINED,(3-MSTCH)
 1000    FORMAT ('BFIELD:BF',I1,I1,I1,I2.2,I1,'.DAT')
C
         INQUIRE (FILE=FILNAM,EXIST=TEST)
         MEXIST = TEST
C
         RETURN
      END
C
C    ****************************** ZEROBM ******************************
C
      SUBROUTINE ZEROBM(LATDIM,LTDIM,I,J,K,L,M,WORK)
C
C   THIS SUBROUTINE ERASES AN INDIVIDUAL B-MATRIX WITHIN THE
C   WORKING B-MATRICES.  THIS IS DONE WHEN AN INDIVIDUAL MATRIX
C   DOES NOT EXIST IN THE OFFLINE B-SUPERMATRIX.
C
      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
C
         DO 10, LAT=1, LATDIM
            DO 20, LT=1, LTDIM
               WORK(LAT,LT,I,J,K,L,M) = 0.0
 20         CONTINUE
 10      CONTINUE
C
         RETURN
      END
```

```
C
C    ****************************** LOADBM ******************************
C
      SUBROUTINE LOADBM(IWANT,LATDIM,LTDIM,I,J,K,L,M,BFNDX,WORK)
C
C    THIS SUBROUTINE LOADS INDIVIDUAL B-MATRICES FROM THE OFFLINE
C    B-SUPERMATRIX INTO THE WORK B-MATRICES.  A CHECK IS MADE TO
C    VERIFY THAT THE CORRECT MATRIX HAS BEEN RETRIEVED.  IF THIS
C    CHECK FAILS THE PROGRAM WILL STOP.
C
      INTEGER BFNDX(2,5)
      CHARACTER*32 FILNAM
C
      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
C
C    THE FILE DEFINITIONS FOR THE B-MATRICES IN THE SUPERMATRIX
C    ARE AS FOLLOWS :
C                    'BFvwxyyz'.
C                         WHERE v IS THE TILT INDEX,
C                               w IS THE STANDOFF INDEX,
C                               x IS THE DST INDEX,
C                              yy IS THE INNER EDGE INDEX,
C                               z IS THE COLLAPSE INDEX.
C
         WRITE (FILNAM,1000) BFNDX(J,2), BFNDX(I,1), BFNDX(L,4),
     1                       BFNDX(K,3),(3-BFNDX(M,5))
 1000    FORMAT ('BFIELD:BF',I1,I1,I1,I2.2,I1,'.DAT')
C
         OPEN (UNIT=99,FILE=FILNAM,ERR=199,FORM='FORMATTED',
     1         ACCESS='SEQUENTIAL',STATUS='OLD')
C
         READ (99,800) IDF1, IDF2, IDF3, IDF4, IDF5, IDF6
  800 FORMAT(3I1,I2.2,I1,I3)
C
         IF ((BFNDX(I,1).NE.IDF2).OR.(BFNDX(J,2).NE.IDF1).OR.
     1       (BFNDX(K,3).NE.IDF4).OR.(BFNDX(L,4).NE.IDF3).OR.
     2       ((3-BFNDX(M,5)).NE.IDF5).OR.(IDF6.NE.201)) THEN
            WRITE (LUERR,*) 'FILE   ',FILNAM,' IS INCORRECT.'
            WRITE (6,*) 'FILE ',FILNAM,' IS INCORRECT.'
            STOP 'INCORRECT B-FIELD. STOPPING IN LOADBM.'
         ENDIF
C
         DO 10, N=1,IWANT
            READ(99,801)((WORK(II,JJ,I,J,K,L,M),II=1,LATDIM),JJ=1,LTDIM)
  801       FORMAT(11E12.4)
 10      CONTINUE
C
         CLOSE (99)
         RETURN
C
 199     CONTINUE
         WRITE(6,*) 'STOPPING IN LOADBM. OPEN FAILED ON ',FILNAM
         STOP 'STOPPING IN LOADBM.  OPEN FAILED ON B-MATRIX.'
      END
C
C    ****************************** TCONV2 ******************************
C
      REAL FUNCTION TCONV2(YR,DD,STRTYR,STRTDD)
C
C    THIS FUNCTION TAKES A START YEAR, START DECIMAL DAY,
C    A CURRENT YEAR, AND A CURRENT DECIMAL DAY.  IT PRODUCES
C    A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
C    THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.
C
```

```
      INTRINSIC INT
C
         TEMP = 0.0
C
         IF (STRTYR.EQ.YR) THEN
            TEMP = (DD - INT(STRTDD)) * 86400.0
         ELSEIF (MOD(INT(STRTYR),4).EQ.0) THEN
            TEMP = (DD + 366.0 - INT(STRTDD)) * 86400.0
         ELSE
            TEMP = (DD + 365.0 - INT(STRTDD)) * 86400.0
         ENDIF
C
         TCONV2 = TEMP
C
         RETURN
      END
C
C    ****************************** TCONV3 ******************************
C
      REAL FUNCTION TCONV3(ITIME,ISTART)
C
C    THIS FUNCTION TAKES A START YEAR, START DAY, A CURRENT YEAR,
C    A CURRENT DAY, AND A CURRENT SECOND..  IT PRODUCES
C    A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
C    THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.
C
      PARAMETER (IYR=1,IDY=2,ISEC=3)
      DIMENSION ITIME(3),ISTART(3)
      INTRINSIC REAL
C
         IF (ISTART(IYR).EQ.ITIME(IYR)) THEN
            TEMP = REAL(ITIME(IDY) - ISTART(IDY))*86400.0 +
     1             REAL(ITIME(ISEC))
         ELSEIF (MOD(ISTART(IYR),4).EQ.0) THEN
            TEMP = REAL(ITIME(IDY) + 366 - ISTART(IDY))*86400.0 +
     1             REAL(ITIME(ISEC))
         ELSE
            TEMP = REAL(ITIME(IDY) + 365 - ISTART(IDY))*86400.0 +
     1             REAL(ITIME(ISEC))
         ENDIF
C
         TCONV3 = TEMP
C
         RETURN
      END
C
C    ***************************** SMOOTH ******************************
C
      SUBROUTINE SMOOTH(DARRY,NDIM,NUMNUM,STARTY,STARTD,NA,XA,YA,IERR)
      DIMENSION  DARRY(7,NDIM),XA(NDIM),YA(NDIM)
      REAL       STARTY,STARTD
      INTEGER    NUMNUM,NDIM,NA,IERR
C
      EXTERNAL TCONV2
C
C     This routine checks and extracts data from DARRY() into XA(),YA().
C     September 10, 1989     Akira NAGAI
C
C     INPUTS:
C        DARRY()         : Source data
C        NDIM            : A Dimension of DARRY()
C        NUMNUM          : Number of source data
C        STARTY, STARTD  : Start year and day
C     OUTPUTS:
```

```
C         NA,XA(),YA()    : # of valid data and valid data pairs
C         IERR            : ERROR CODE
C
C      September 10, 1989
C
C      Time conversion
       DO 5,I=1,NUMNUM
          XA(I) = TCONV2(DARRY(2,I),DARRY(3,I),STARTY,STARTD)
C         WRITE (6,*) 'SMOOTH --',DARRY(2,I),DARRY(3,I),STARTD,XA(I)
          YA(I) = DARRY(1,I)
 5     CONTINUE
       NA = NUMNUM

C      Check Time-tag.  Shift-up if same.
       IERR=0
       I=0
 10    CONTINUE
       I=I+1
       IF (I.GE.NA) RETURN
       IF (XA(I).GT.XA(I+1)) THEN
          WRITE(6,*) 'SERIOUS ERROR: TIME-TAG REVERSE ORDER IN DARRY()'
          WRITE(6,*) 'X(I) = ',XA(I),'  X(I+1) = ',XA(I+1),'  I = ',I
          IERR = -1
          RETURN
       ELSE
          IF (XA(I).EQ.XA(I+1)) THEN
             WRITE(6,*) 'WARNING: SAME TIME-TAG EXISTS IN DARRY().'
             WRITE(6,*) 'SECOND DATA IGNORED.'
             DO 20,K=I+1,NA-1
                XA(K)=XA(K+1)
                YA(K)=YA(K+1)
 20          CONTINUE
             NA = NA - 1
             I = I - 1
          END IF
       END IF
       GOTO 10
       END
C
C    ***************************** DTNTRP *********************************
C
       SUBROUTINE  DTNTRP(X,NA,XA,YA,Y,DYDX,DELTA,IERR)
       INTEGER  NA,IERR,I
       REAL     X,XA(*),YA(*),XD,Y,DYDX,DELTA
C
C      This function returns interpolation values for the
C      N data pairs of (XA(i),YA(i)) , where i=1..NA.
C
C      INPUTS:
C         X  : Given value of X
C         NA : Number of data points.
C         XA : Data   XA(1) to XA(NA)
C         YA : Data   YA(1) to YA(NA)
C
C      OUTPUTS:
C         Y    : Interpolated value of Y.
C         DYDX: Interpolated value of dY/dX.
C         DELTA : Gap of X.
C         IERR: Error code.
C
C      September 08, 1989
C
       IF (X.GE.XA(NA)) THEN
          IERR=-1
```

```
         IF (X.EQ.XA(NA)) IERR=0
         Y = YA(NA)
         DYDX = ( YA(NA)-YA(NA-1) ) / ( XA(NA)-XA(NA-1) )
         DELTA = X - XA(NA)
      ELSE IF (X.LT.XA(1)) THEN
         IERR=-1
         Y = YA(1)
         DYDX = ( YA(2)-YA(1) ) / ( XA(2)-XA(1) )
         DELTA = XA(1) - X
      ELSE

C     Scan to find the position of X.
C     Maybe binary search would be better.
         I = NA
10       CONTINUE
            XD = X - XA(I)
         IF (XD.GT.0.0) GOTO 20
            I = I - 1
         GOTO 10

20       CONTINUE
         IERR = 0
         DELTA2= XA(I+1)-XA(I)
         DELTA = MIN(X-XA(I),XA(I+1)-X)
         DYDX = (YA(I+1)-YA(I)) / DELTA2
         Y = DYDX * XD + YA(I)
      ENDIF
      RETURN
      END
C
C    ***************************** DTXIPT ****************************
C
      SUBROUTINE  DTXIPT(X,NA,XA,YA,Y,DYDX,DELTA,IERR)
C
      INTEGER  NA,IERR,I
      REAL     X,XA(*),YA(*),XD,Y,DYDX,DELTA
C
C     This function returns interpolation values for XIPATT
C
C     INPUTS:
C        X  : Given value of X
C        NA : Number of data points.
C        XA : Data   XA(1) to XA(NA)
C        YA : Data   YA(1) to YA(NA)
C
C     OUTPUTS:
C        Y    : Interpolated value of Y.
C        DYDX: Interpolated value of dY/dX.
C        DELTA : Gap of X.
C        IERR: Error code.
C
C     September 08, 1989
C
      IF (X.GT.XA(NA)) THEN
         IERR=-1
         IF (X.EQ.XA(NA)) IERR=0
         Y = YA(NA)
         DELTA = X - XA(NA)
      ELSE IF (X.LE.XA(1)) THEN
         IERR=-1
         IF (X.EQ.XA(1)) IERR=0
         Y = YA(1)
         DELTA = XA(1) - X
      ELSE
```

```
C     Scan to find the position of X.
         I = NA
10       CONTINUE
            XD = X - XA(I)
         IF (XD.GE.0.0) GOTO 20
            I = I - 1
         GOTO 10

20       CONTINUE
         IERR = 0
         DELTA = MIN(X-XA(I),XA(I+1)-X)
         A=X-XA(I)
         B=XA(I+1)-X

         IF (A.LT.B) THEN
            Y = YA(I)
         ELSE
            Y = YA(I+1)
         END IF
      ENDIF

      RETURN
      END
C*******************************************************   STDOFF   ****
C
      SUBROUTINE STNDOF(VELOC,DENS,STAND)
C
C                    ANALYTIC CALCULATION OF THE STAND-OFF DISTANCE *
C
C                    FORMULA ACCORD. TO ALPBACH - PAPER (1979)
C
C                    B0  = MAGNETIC FIELD AT THE EQUATOR IN (GAUSS)
C                    FAK = REFLECTION FACTOR FOR SOLAR-WIND PATICLES
C                    F   = GEOMETRY FAKTOR OF THE MAGNETOPAUSE
C                  PSOL  = SOLARWIND-PRESS IN (1.E-08 DYN/(CM**2))
C                  PEFF  = SOLARWIND-PRESS IN (1.E-16 DYN/(CM**2)
C
C
C * * * INPUT PARAMTERS:     VELOC  SOL WIND VELOCITY IN [KM/SEC]
C                            DENS   SOL WIND DESITY IN [1/CM]**3
C
C * * * OUTPUT PARAMETERS:   BSTAG  MAGNETIC FIELD AT SUBSOLAR POINT IN [
C                            STAND  STAND OFF DISTANCE IN [EARTH RADII]
C
C
C * * CALCULATION OF HELIUM PERCENTAGE IN THE SOLAR WIND    * * * * *
C
C     HELIUM  =  HELIUM-PROZENT
C
      HELIUM = 0.00
C
      PERC1 = HELIUM/100.
      PERC2 = 1.0 - PERC1
C
      DENGES = DENS * (PERC2 + 4.*PERC1)
C
      V0 = VELOC
C
      FAK = 1.0
      PROTMA = 1.6726 * 1.E-24
      VAU    = V0     * 1.E+05
      PI   = ACOS(-1.00000)
C
```

```
      DRUCK = 8.*PI * FAK * DENGES * PROTMA * VAU * VAU
      DRUCK = DRUCK * 1.E+10
      DRUCK = SQRT(DRUCK)
C
      FELD = DRUCK
C
      PSOL = DRUCK * 1.E-05
      PSOL = PSOL * PSOL
      PSOL = PSOL/(8.*PI)
      PSOL = PSOL * 1.E+08
C
C
      PI = ACOS(-1.00000)
C     B0 = 0.23
      B0 = 0.311829
C
      PGES = PSOL * (PERC2 + 4.*PERC1)
C
C
C * * IN THIS SPECIAL VERSION OF THE ROUTINE STDOFF, THE MAGNETOSPHERE I
C                                     ASSUMED TO BE COMPLETELY CLOSED.
C
C
      BZIMF = 0.00
      CD    = 0.00
      CIMF  = 0.00
C
      BZIMF = BZIMF * 1.E-05
C
      F = 1.16000
      FAK = 1.0
      FQK = F * F / FAK
C
      PEFF = 2.*PI * PGES * 1.E-08
      EX   = 1./6.
C
      R0 = (FQK * B0*B0*(1.-CD)*(1.-CD)/ PEFF)**EX
      R0 = R0 * (1. + (EX*CIMF*BZIMF)/(SQRT(PEFF)))
C
      FELD = PGES * 1.E-08
      FELD = FELD * 8.*PI
      FELD = SQRT(FELD)
      FELD = FELD * 1.E+05
C
      BSTAG = FELD
      STAND = R0
C
C
      RETURN
      END
      SUBROUTINE FLXNIT(IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,
     2                  FLXNRG,FLXKP,FLXMAT,THRMAT)
C
C  VERSION 1.00                        DATE: 10.04.89
C
C  PURPOSE: SUBROUTINE TO CALCULATE KP-BASED PARTICLE NUMBER FLUX
C           IN UNITS OF PARTICLES/CM**2/S/SR/EV AS FCN OF R, ENERGY,
C           AND KP.
C
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)
      DIMENSION THRMAT(IRDIM,NRGDIM,KPDIM)
      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)
      DIMENSION THR(4)
C
```

```
C
      DO 10 IKP=1,KPDIM
         FLXKP(IKP)=FLOAT(IKP-1)
   10 CONTINUE
C
      DO 20 INRG=1,NRGDIM
         FLXNRG(INRG)=FLOAT(INRG+3)/4.
   20 CONTINUE
C
      FLXR(1)=ALOG10(3.)
      FLXR(2)=ALOG10(4.)
      FLXR(3)=ALOG10(6.6)
      FLXR(4)=ALOG10(13.)
C
C
      DO 90 ISP=1,ISPDIM
         ISPP=ISP
         DO 100 IKP=1,KPDIM
            DO 110 INRG=1,NRGDIM
              CALL FLXCAL(ISPDIM,KPDIM,ISPP,FLXKP(IKP),FLXNRG(INRG),
     2                    FLX3,FLX4,FLX6,FLX13)
              DO 120 IR=1,IRDIM
                 FLXMAT(IR,INRG,IKP,ISP)=FLXTRP(FLXR(IR),FLX3,FLX4,
     2                                     FLX6,FLX13)
  120          CONTINUE
  110       CONTINUE
  100    CONTINUE
   90 CONTINUE
C
      DO 200 IKP=1,KPDIM
         IIKP=IKP
         DO 210 INRG=1,NRGDIM
            IINRG=INRG
            CALL THRCAL(FLXKP(IKP),FLXNRG(INRG),IINRG,IIKP,IRDIM,
     2                  NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,THR)
            DO 220 IR=1,IRDIM
               THRMAT(IR,INRG,IKP)=THR(IR)
  220       CONTINUE
  210    CONTINUE
  200 CONTINUE
C
C  CYCLE THROUGH KP VALUES AND PUT MIN FLUX IN KPDIM+1
C                                  MAX FLUX IN KPDIM+2
C
      DO 300 ISP=1,ISPDIM
         DO 310 INRG=1,NRGDIM
            DO 320 IR=1,IRDIM
               FLXMIN=1.E20
               FLXMAX=-1.E20
               DO 330 IKP=1,KPDIM
                  FLXMIN=MIN(FLXMAT(IR,INRG,IKP,ISP),FLXMIN)
                  FLXMAX=MAX(FLXMAT(IR,INRG,IKP,ISP),FLXMAX)
  330          CONTINUE
C
               FLXMAT(IR,INRG,KPDIM+1,ISP)=FLXMIN
               FLXMAT(IR,INRG,KPDIM+2,ISP)=FLXMAX
  320       CONTINUE
  310    CONTINUE
  300 CONTINUE
C
C
C
      RETURN
      END
```

```
      SUBROUTINE FLXCAL(ISPDIM,KPDIM,ISPP,FKP,FNRGLG,FLX3,FLX4,FLX6,
     2                  FLX13)
C
C  VERSION 1.00                      DATE: 05.02.90
C
C  PURPOSE: COMPUTE KP-DEPENDENT FLUX (PARTICLES/CM**2/S/SR/EV) AT
C           L=3, 4, 6.6, AND 13 FOR ELECTRONS OR IONS OF ENERGY
C           FNRG (EV) AND FOR KP=FKP CONDITIONS.  ISPP GIVES THE
C           MASS SPECIES OF THE FLUX TO BE CALCULATED.
C           IF ISPP=1  THEN ELECTRON FLUX VALUES ARE RETURNED
C           IF ISPP=2  THEN H+ FLUX VALUES ARE RETURNED
C           IF ISPP=3  THEN O+ FLUX VALUES ARE RETURNED
C
C
C  REF: SECTION 2.5 OF THE FINAL REPORT FOR
C       CONTRACT #F19628-87-K-0001
C
      PARAMETER (ISPSIZ=3,IKPSIZ=7,KAPSIZ=2)
      PARAMETER(NKPSIZ=3,NSPSIZ=2)
C
      DIMENSION DENGEO(KAPSIZ,IKPSIZ),EGEO(KAPSIZ,IKPSIZ)
      DIMENSION DIONG(KAPSIZ,IKPSIZ),EIONG(KAPSIZ,IKPSIZ)
C
      DIMENSION DEN4(NKPSIZ,KAPSIZ,NSPSIZ),DEN3(NKPSIZ,KAPSIZ,NSPSIZ)
      DIMENSION EZ43(NKPSIZ,KAPSIZ,NSPSIZ),RKAP43(NKPSIZ,KAPSIZ,NSPSIZ)
      DIMENSION FKAP43(NKPSIZ,KAPSIZ,NSPSIZ),FLX43(NKPSIZ)
C
C
      DATA DENGEO /0.5697, .004926,  0.6034,.2261,  0.6292, .2583,
     2             0.7002, .3418,  0.8135, .4602,  1.0347, .6446,
     3             1.4468, .8505/
C
      DATA EGEO /31.036, 21000., 40.803, 4832.4, 48.670, 4814.0,
     2           71.680, 4797.8, 111.119, 4816.0, 192.704,4894.3,
     3           342.712, 5075.2/
C
      DATA DIONG /0.602,0.460,0.661,0.457,
     2            0.712,0.495,0.813,0.524,
     3            0.966,0.539,0.700,0.517,
     4            1.652,0.453/
C
      DATA EIONG /  180.2,25408.,  258.4,26837.,
     2              335.4,27902.,  604.8,30723.,
     3             1144.2,34972., 2315.4,42809.,
     4             4182.4,57163./
C
      DATA DEN4/0.10,0.20,0.00,0.375,1.90,2.30,
     2          0.43,0.18,0.00,0.00,1.50,5.00/
C
      DATA DEN3/0.032,0.06,0.00,0.12,0.60,6.30,
     2          0.06,0.018,0.0,0.00,0.15,11.8/
C
      DATA EZ43/3.00,15.0,15.00,200.,40.0,60.0,
     2          5.00,5.00,5.00,20.0,20.0,30.0/
C
      DATA FKAP43/.9359421,.9619483,.9619483,
     2            .9619483,.9359421,.9619483,
     3            .9027036,.9619483,.9619483,
     4            .9359421,.9359421,.9619483/
C
      DATA RKAP43/6.00,10.0,10.00,10.0,6.00,10.0,
     2            4.00,10.0,10.00,6.00,6.00,10.0/
C
C
```

```
      IF(ISPSIZ.NE.ISPDIM) THEN
         WRITE(6,*) 'DIMENSION ERROR IN SUBROUTINE FLXNIT'
         WRITE(6,*) 'ISPSIZ IS NOT EQUAL TO ISPDIM'
         WRITE(6,*) 'ISPSIZ=',ISPSIZ,'ISPDIM=',ISPDIM
         WRITE(6,*) 'STOPPING PROGRAM IN FLXNIT'
         STOP
      END IF
C
      IF(IKPSIZ.NE.KPDIM) THEN
         WRITE(6,*) 'DIMENSION ERROR IN SUBROUTINE FLXNIT'
         WRITE(6,*) 'IKPSIZ IS NOT EQUAL TO KPDIM'
         WRITE(6,*) 'IKPSIZ=',ISPSIZ,'KPDIM=',KPDIM
         WRITE(6,*) 'STOPPING PROGRAM IN FLXNIT'
         STOP
      END IF
C
      FNRG=10.**FNRGLG
C
      IF(ISPP.EQ.1) THEN
C        CALCULATE ELECTRON FLUX VALUES
         RKAP=6.
         FKAP=.9359421
C
C  CALCULATE PLASMA SHEET FLUX, FLX13
         DENSTY=(.4*(FKP-1.)+0.5*(5.-FKP))/4.
         EZERO=.2885*(FKP-1.)+.15625*(5.-FKP)
         FLX13=1.68E5*FKAP*DENSTY*(FNRG*1.E-3/EZERO)/SQRT(EZERO)/
     2         (1.+(FNRG*1.E-3)/RKAP/EZERO)**(RKAP+1.)
C
C
C  CALCULATE GEOSYNCHRONOUS FLUX, FLX6
         IKP=NINT(FKP+1.)
         FLX6=0.
         DO 10 N=1,2
C           WRITE(6,*) 'IKP=',IKP,' N=',N,' FNRG=',FNRG
C           WRITE(6,*) 'DENSITY=',DENGEO(N,IKP),' EZERO=',EGEO(N,IKP)
C           WRITE(6,*) 'FKAP=',FKAP,' RKAP=',RKAP
            FLX6=FLX6+1.68E5*FKAP*DENGEO(N,IKP)*(FNRG/EGEO(N,IKP))/
     2           SQRT(EGEO(N,IKP)*1.E-3)/
     3           (1.+MIN(FNRG/RKAP/EGEO(N,IKP),1.E5))**(RKAP+1.)
C           WRITE(6,*) 'FLX6=',FLX6
   10    CONTINUE
C
C
C  CALCULATE FLUX AT L=4, FLX4
         FLX4=1.106E1*(FNRG*1.E-3)*(10.**((FKP-2.)/6.))/
     2           (1.+FNRG*1.E-3/32.)**3
C        WRITE(6,*) 'FLX4=',FLX4
C
C
C  CALCULATE FLUX AT L=3, FLX3
         FLX3=4.795E0*(FNRG*1.E-3)*(10.**((FKP-2.)/6.))/
     2           (1.+FNRG*1.E-3/65.8)**3.8
C
C
C
      ELSE
C         CALCULATE H+ OR O+ FLUXES
C
C
C  PLASMA SHEET FLUX, FLX13
         DENSTY=(.4*(FKP-1.)+0.5*(5.-FKP))/4.
         EZERO=2.25*(FKP-1.)+1.219*(5.-FKP)
C  EZERO IN KEV
```

```
      RKAP=6.
      FKAP=.9359421
C
      IF(FKP.GE.1.) THEN
         RATOH=(0.6*(FKP-1.)+0.06*(5.-FKP))/4.
      ELSE
         RATOH=0.06
      END IF
C
      IF (ISPP.EQ.2) THEN
         DENION=DENSTY/(1.+RATOH)
         AWT=1.
      ELSE
         DENION=DENSTY*RATOH/(1.+RATOH)
         AWT=16.
      END IF
C
      FLX13=3.93E3*FKAP*DENION*(FNRG*1.E-3/EZERO)/SQRT(AWT*EZERO)/
     2             (1.+(FNRG*1.E-3)/RKAP/EZERO)**(RKAP+1.)
C
C
C CALCULATE GEOSYNCHRONOUS ION FLUX, FLX6
      RKAP1=10.
      RKAP2=4.
      FKAP1=.96194826
      FKAP2=.9027036
C
      RATOH1=(0.7*(FKP-1.)+0.4*(5.-FKP))/4.
C
      IF(FKP.GE.1.) THEN
         RATOH2=(0.154*(5.-FKP)+0.767*(FKP-1.))/4.
      ELSE
         RATOH2=0.154
      END IF
C
      IKP=NINT(FKP+1.)
      FLX6=0.
C
      DO 20 N=1,2
C
         IF(N.EQ.1) THEN
            RKAP=RKAP1
            FKAP=FKAP1
            RATOH=RATOH1
         ELSE
            RKAP=RKAP2
            FKAP=FKAP2
            RATOH=RATOH2
         END IF
C
         IF(ISPP.EQ.2) THEN
C           H+ IONS
            DENION=DIONG(N,IKP)/(1.+RATOH)
            AWT=1.
         ELSE
C           O+ IONS
            DENION=DIONG(N,IKP)*RATOH/(1.+RATOH)
            AWT=16.
         END IF
C
         FLX6=FLX6+3.93E3*FKAP*DENION*(FNRG/EIONG(N,IKP))/
     2                 SQRT(AWT*EIONG(N,IKP)*1.E-3)/
     3                 (1.+FNRG/RKAP/EIONG(N,IKP))**(RKAP+1.)
C
```

```
   20     CONTINUE
C
C
C  CALCULATE ION FLUX AT L=4, FLX4
C    NOTE: THIS ALGORITHM IS A LITTLE WASTEFUL OF TIME IN THE INTEREST
C          OF SAVING PROGRAM COMPLEXITY.
C
          ISPNDX=ISPP-1
          DO 30 NKP=1,NKPSIZ
C
C  NKP=1 IS ASSOCIATED WITH KP=0
C  NKP=2 IS ASSOCIATED WITH KP=5+
C  NKP=3 IS ASSOCIATED WITH KP=9-
C
               FLX43(NKP)=0.
C
               DO 40 N=1,2
                    FLX43(NKP)=FLX43(NKP)+3.93E3*FKAP43(NKP,N,ISPNDX)*
     2                    DEN4(NKP,N,ISPNDX)*(FNRG*1.E-3/
     3                    EZ43(NKP,N,ISPNDX))/
     4                    SQRT(AWT*EZ43(NKP,N,ISPNDX))/
     5                    (1.+(FNRG*1.E-3)/RKAP43(NKP,N,ISPNDX)/
     6                   EZ43(NKP,N,ISPNDX))**(RKAP43(NKP,N,ISPNDX)+1.)
C
   40          CONTINUE
C
   30     CONTINUE
C
          IF(FKP.LE.5.34) THEN
               FLXLOG=ALOG10(FLX43(1))+ALOG10(FLX43(2)/FLX43(1))*
     2                             (FKP/5.34)
          ELSE IF(FKP.GT.5.34.AND.FKP.LE.8.67) THEN
               FLXLOG=ALOG10(FLX43(2))+ALOG10(FLX43(3)/FLX43(2))*
     2                             ((FKP-5.34)/(8.67-5.34))
          ELSE
               FLXLOG=ALOG10(FLX43(3))
          END IF
C
          FLX4=10.**FLXLOG
C
C
C
C
C  CALCULATE ION FLUX AT L=3, FLX3
C    NOTE: THIS ALGORITHM IS A LITTLE WASTEFUL OF TIME IN THE INTEREST
C          OF SAVING PROGRAM COMPLEXITY.
C
          ISPNDX=ISPP-1
          DO 50 NKP=1,NKPSIZ
C
C  NKP=1 IS ASSOCIATED WITH KP=0
C  NKP=2 IS ASSOCIATED WITH KP=5+
C  NKP=3 IS ASSOCIATED WITH KP=9-
C
               FLX43(NKP)=0.
C
               DO 60 N=1,2
                    FLX43(NKP)=FLX43(NKP)+3.93E3*FKAP43(NKP,N,ISPNDX)*
     2                    DEN3(NKP,N,ISPNDX)*(FNRG*1.E-3/
     3                    EZ43(NKP,N,ISPNDX))/
     4                    SQRT(AWT*EZ43(NKP,N,ISPNDX))/
     5                    (1.+(FNRG*1.E-3)/RKAP43(NKP,N,ISPNDX)/
     6                   EZ43(NKP,N,ISPNDX))**(RKAP43(NKP,N,ISPNDX)+1.)
C
```

```
   60        CONTINUE
C
   50     CONTINUE
C
          IF(FKP.LE.5.34) THEN
             FLXLOG=ALOG10(FLX43(1))+ALOG10(FLX43(2)/FLX43(1))*
     2                               (FKP/5.34)
          ELSE IF(FKP.GT.5.34.AND.FKP.LE.8.67) THEN
             FLXLOG=ALOG10(FLX43(2))+ALOG10(FLX43(3)/FLX43(2))*
     2                               ((FKP-5.34)/(8.67-5.34))
          ELSE
             FLXLOG=ALOG10(FLX43(3))
          END IF
C
          FLX3=10.**FLXLOG
C
C
       END IF
C
C
       RETURN
       END
       FUNCTION FLXTRP(RLOG,FLX3,FLX4,FLX6,FLX13)
C
C  VERSION 1.00                       DATE: 10.04.89
C
C  PURPOSE: FUNCTION SUBPROGRAM TO RETURN FLUX VALUE INTERPOLATED
C                BETWEEN VALUES AT L=3, 4, 6.6, AND 13 RE.
C
C
       FR=10.**RLOG
       IF(FR.LE.4.) THEN
          FLXLOG=(ALOG10(FR/3.)*ALOG10(FLX4)+ALOG10(4./FR)*ALOG10(FLX3))
     2           /ALOG10(4./3.)
       ELSE IF(FR.GT.4..AND.FR.LE.6.6) THEN
          FLXLOG=(ALOG10(FR/4.)*ALOG10(FLX6)+ALOG10(6.6/FR)*ALOG10(FLX4))
     2           /ALOG10(6.6/4.)
       ELSE
          FLXLOG=ALOG10(FLX13)+ALOG10(13./FR)
       END IF
C
       FLXTRP=FLXLOG
C
C
       RETURN
       END
       FUNCTION FLXVAL(ISP,FKP,RR,PP,ENRG,IRDIM,NRGDIM,KPDIM,KPPLUS,
     2                 ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP)
C
C  VERSION 1.00                       DATE: 05.08.90
C                                           12.31.89
C                                           05.30.90
C
C  PURPOSE: CALCULATE KP-DEPENDENT FLUX VALUE AT R=RR, FOR PARTICLES OF
C                SPECIES ISP, OF ENERGY ENRG AND KP=FKP CONDITIONS BY
C                INTERPOLATING FLXMAT ARRAY.
C
C  PROGRAMMER: R.W. SPIRO
C
       DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)
       DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)
C
C  DETERMINE INTERPOLATION INDICES
C
```

```
      IF(FKP.GE.0.) THEN
         BKP=FKP+1.
         IF(BKP.GT.FLOAT(KPDIM)) BKP=FLOAT(KPDIM)
      ELSE
         BKP=ABS(FKP)
      END IF
C
      BNRG=4.*ALOG10(ENRG)-3.
C
      RLOG=ALOG10(RR)
C
      IF(RLOG.GT.FLXR(IRDIM)) THEN
         BR=IRDIM
         FLXLOG=G3NTRP(FLXMAT(1,1,1,ISP),IRDIM,NRGDIM,KPPLUS,BR,BNRG,
     2                                      BKP) + ALOG10(13./RR)
C
      ELSE
C
         DO 10 IR=1,IRDIM-1
            IF(RLOG.LE.FLXR(IR+1)) THEN
               BR=FLOAT(IR)+(RLOG-FLXR(IR))/(FLXR(IR+1)-FLXR(IR))
               GO TO 20
            END IF
   10    CONTINUE
C
C
   20    CONTINUE
C        WRITE(6,*) 'FKP,RR,ENRG',FKP,RR,ENRG
C        WRITE(6,*) 'BKP,BR,BNRG',BKP,BR,BNRG
C
        FLXLOG=G3NTRP(FLXMAT(1,1,1,ISP),IRDIM,NRGDIM,KPPLUS,BR,BNRG,
     2                 BKP)
      END IF
C
      FLXVAL=10.**FLXLOG
C     WRITE(6,*) '      FLXLOG=',FLXLOG,'  FLXVAL=',FLXVAL
C
      RETURN
      END
      SUBROUTINE EFIELD(KPATT,PCP,DEQDT,ASOUTH,BSOUTH,DXS,DYS,ANORTH,
     2                  BNORTH,DXN,DYN,COLAT,ALOCT,MODE,LATDIM,LTDIM,
     3                  JWRAP,ITMDIM,ITMCUR,ICNTRL,ITOP,V,VNORTH,
     4                  VSOUTH)
C
C PURPOSE:  SUBROUTINE TO RETURN VALUES OF THE POTENTIAL ON THE
C           NORTHERN HEMISPHERE GRID (VNORTH), ON THE SOUTHERN
C           HEMISPHERE GRID (VSOUTH), AND THE AVERAGE OF THE TWO
C           HEMISPHERES (V).  THE POTENTIAL DISTRIBUTION GIVEN BY
C           V IS THE ACTUAL DISTRIBUTION USED TO FOLLOW PARTICLE
C           TRACES.  THIS ROUTINE CALLS SUBROUTINE EMODEL TO ACTUALLY
C           COMPUTE THE POTENTIAL DISTRIBUTIONS.
C
C NOTE:  KPATT GIVES THE UTD DEFINED PATTERN TYPE WITH 0 CORRESPONDING
C        TO NORTHWARD IMF CASES; 1 TO BZ SOUTH WITH STRONGER FLOW ON
C        THE DAWN SIDE OF THE POLAR CAP; 2 TO BZ SOUTH WITH SYMMETRIC
C        FLOW; AND 3 TO BZ SOUTH WITH STRONGER FLOW ON THE DUSK SIDE.
C        THIS SUBROUTINE CALLS EMODEL AFTER TRANSLATING THE PATTERN
C        TYPE TO THE INTERNAL EMODEL CONVENTION AS GIVEN BELOW:
C
C   GENERAL CALL PARAMETERS FOR SUBROUTINE EMODEL
C
C     IPATT = HEPPNER-MAYNARD PATTERN NO.  MUST BE SPECIFIED AS AN
C             INTEGER BETWEEN 1 AND 7.
C             IPATT=1 MEANS PATTERN A, FOR BZ SOUTH, BY=0.
```

```
C               IPATT=2 MEANS PATTERN BC, FOR BZ SOUTH, BYNORTH>0               MSM67200
C               IPATT=3 MEANS PATTERN DE, FOR BZ SOUTH,BYNORTH<0                MSM67210
C               IPATT=4 MEANS PATTERN BCP,TWISTED BC, FOR BZ>0                  MSM67220
C               IPATT=5 MEANS PATTERN BCPP,TWISTED BC,BZ STRONG>0               MSM67230
C               IPATT=6 MEANS PATT. DEP, TWISTED DE,BZ WEAK >0                  MSM67240
C               IPATT=7 MEANS PATT. DEPP, TWISTED DE,BZ STRONG>0                MSM67250
C                                                                               MSM67260
C  FOR THE PRESENT, NORTHWARD IMF CASES ARE ALL TREATED AS IPATT=1              MSM67270
C                                                                               MSM67280
C  PROGRAMMER:  R.W. SPIRO                   LAST UPDATE: 11-17-89              MSM67290
C                                                                               MSM67300
C                                                                               MSM67310
      DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM)                         MSM67320
      DIMENSION V(LATDIM,LTDIM,ITMDIM),VNORTH(LATDIM,LTDIM)                     MSM67330
      DIMENSION VSOUTH(LATDIM,LTDIM)                                            MSM67340
      DIMENSION ASOUTH(3,ITMDIM),BSOUTH(3,ITMDIM)                               MSM67350
      DIMENSION DXS(3,ITMDIM),DYS(3,ITMDIM)                                     MSM67360
      DIMENSION ANORTH(3,ITMDIM),BNORTH(3,ITMDIM)                               MSM67370
      DIMENSION DXN(3,ITMDIM),DYN(3,ITMDIM)                                     MSM67380
C                                                                               MSM67390
C                                                                               MSM67400
      IF(KPATT.EQ.0) THEN                                                       MSM67410
C  NORTHWARD BZ CASE                                                            MSM67420
         IPATT=1                                                                MSM67430
         CALL EMODEL(IPATT,PCP,DEQDT,ASOUTH(1,ITMCUR),BSOUTH(1,ITMCUR),         MSM67440
     2               DXS(1,ITMCUR),DYS(1,ITMCUR),COLAT,ALOCT,                   MSM67450
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VSOUTH)                MSM67460
         CALL EMODEL(IPATT,PCP,DEQDT,ANORTH(1,ITMCUR),BNORTH(1,ITMCUR),         MSM67470
     2               DXN(1,ITMCUR),DYN(1,ITMCUR),COLAT,ALOCT,                   MSM67480
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VNORTH)                MSM67490
C                                                                               MSM67500
      ELSE IF(KPATT.EQ.-1.OR.KPATT.EQ.3) THEN                                   MSM67510
C  CASE OF BZ SOUTH, BY<0                                                       MSM67520
C                                                                               MSM67530
         IPATT=2                                                                MSM67540
         CALL EMODEL(IPATT,PCP,DEQDT,ASOUTH(1,ITMCUR),BSOUTH(1,ITMCUR),         MSM67550
     2               DXS(1,ITMCUR),DYS(1,ITMCUR),COLAT,ALOCT,                   MSM67560
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VSOUTH)                MSM67570
C                                                                               MSM67580
         IPATT=3                                                                MSM67590
         CALL EMODEL(IPATT,PCP,DEQDT,ANORTH(1,ITMCUR),BNORTH(1,ITMCUR),         MSM67600
     2               DXN(1,ITMCUR),DYN(1,ITMCUR),COLAT,ALOCT,                   MSM67610
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VNORTH)                MSM67620
C                                                                               MSM67630
      ELSE IF(KPATT.EQ.1.OR.KPATT.EQ.-3) THEN                                   MSM67640
C  CASE OF BZ SOUTH, BY>0                                                       MSM67650
C                                                                               MSM67660
         IPATT=3                                                                MSM67670
         CALL EMODEL(IPATT,PCP,DEQDT,ASOUTH(1,ITMCUR),BSOUTH(1,ITMCUR),         MSM67680
     2               DXS(1,ITMCUR),DYS(1,ITMCUR),COLAT,ALOCT,                   MSM67690
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VSOUTH)                MSM67700
C                                                                               MSM67710
         IPATT=2                                                                MSM67720
         CALL EMODEL(IPATT,PCP,DEQDT,ANORTH(1,ITMCUR),BNORTH(1,ITMCUR),         MSM67730
     2               DXN(1,ITMCUR),DYN(1,ITMCUR),COLAT,ALOCT,                   MSM67740
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VNORTH)                MSM67750
C                                                                               MSM67760
      ELSE                                                                      MSM67770
C  CASE OF BZ SOUTH, SYMMETRIC FLOW                                             MSM67780
C                                                                               MSM67790
         IPATT=1                                                                MSM67800
         CALL EMODEL(IPATT,PCP,DEQDT,ASOUTH(1,ITMCUR),BSOUTH(1,ITMCUR),         MSM67810
     2               DXS(1,ITMCUR),DYS(1,ITMCUR),COLAT,ALOCT,                   MSM67820
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VSOUTH)                MSM67830
```

```
C
         CALL EMODEL(IPATT,PCP,DEQDT,ANORTH(1,ITMCUR),BNORTH(1,ITMCUR),
     2               DXN(1,ITMCUR),DYN(1,ITMCUR),COLAT,ALOCT,
     3               MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VNORTH)
C
      END IF
C
C  COMPUTE V=POTENTIAL USED IN TRACING PARTICLE TRAJECTORIES=AVG OF
C            VNORTH AND VSOUTH
C
      DO 100 J=1,LTDIM
         DO 110 I=1,LATDIM
            V(I,J,ITMCUR)=(VSOUTH(I,J)+VNORTH(I,J))/2.
C
  110    CONTINUE
  100 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE PWRCAL(LATDIM,LTDIM,ISPDIM,ITMDIM,ITMCUR,FKP,FLXSUM,
     2                  EAVG,COLAT,ALOCT,ALPHA,BETA,BNDLOC,A,B,DX,DY)
C
C  PURPOSE:  ESTIMATE PRECIPITATING ELECTRON ENERGY FLUX POLEWARD OF
C          THE MAIN MSM MODELING REGION BY COMPARING FLUX WITHIN
C          MODELING REGION WITH HARDY STATISTICAL VALUES.  THIS ROUTINE
C          IS BASED ON SUBROUTINE PWRCAL IN THE RCM SOURCE CODE; THE
C          RATIONALE FOR THE METHOD IS GIVEN IN AN R.A. WOLF DOCUMENT
C          'PROPOSED PROCEDURE FOR ESTIMATING PRECIPITATING ENERGY
C          FLUX AND AVERAGE ENERGY POLEWARD OF RCM'S MAIN MODELING
C          REGION' DATED 02-27-89.
C
C  PROGRAMMER: R.W. SPIRO                LAST UPDATE: 11-29-89
C
C  CALLING PARAMETERS:
C        LATDIM          LATITUDINAL DIMENSION
C        LTDIM           LOCAL TIME DIMENSION
C        ISPDIM          MASS SPECIES DIMENSION
C        ITMDIM          TIME LABEL DIMENSION
C        ITMCUR          CURRENT TIME LABEL
C        FKP             FLOATING PT KP VALUE
C        FLXSUM          PRECIPITATING ENERGY FLUX ARRAY IN UNITS OF
C                        ERG/CM**2/S
C        EAVG            AVERAGE ENERGY ARRAY
C        COLAT           COLATITUDE ARRAY (RADIANS)
C        ALOCT           LOCAL TIME ANGLE ARRAY (RADIANS EASTWARD FROM
C                        LOCAL NOON)
C        ALPHA           LATITUDINAL GRID SPACING VECTOR
C        BETA            LOCAL TIME GRID SPACING VECTOR
C        BNDLOC          OUTER BNDY OF DETAILED MSM CALCULATION
C        A,B,DX,DY       PARAMETERS SPECIFYING ELECTRIC FIELD BNDY
C                        ELLIPSES
C
C

      PARAMETER(JDIM=51)

      COMMON /GRID/ DLAM,DPSI,RI,JWRAP
C
      DIMENSION FLXSUM(LATDIM,LTDIM,ISPDIM),EAVG(LATDIM,LTDIM,ISPDIM)
      DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM)
      DIMENSION ALPHA(LATDIM),BETA(LATDIM),BNDLOC(LTDIM,ITMDIM)
      DIMENSION A(3),B(3),DX(3),DY(3)
C
```

```
      DIMENSION XMLT(26),PWRKP(26,7),PTOT(JDIM),PWRTOT(JDIM)
      DIMENSION FRACT(JDIM),RENORM(JDIM),PWRMDL(JDIM)

C PWRMDL(J)=TOTAL PRECIPITATED POWER WITHIN MODEL REGION PER J GRID LINE
C PTOT(J)=OBSERVED POWER PER HOUR OF LOCAL TIME*D(LT)/DJ AT JGRID FOR
C          GIVEN KP VALUE
C
C PWRTOT(J)=OBSERVED POWER CORRECTED FOR OUR MODELING REGION
C          =PTOT(J)*(1.-FRACT(J))
C
C RENORM(J)=PWRMDL(J)/PWRTOT(J)
C
C

      DATA XMLT/11.5,12.5,13.5,14.5,15.5,16.5,17.5,18.5,
     $          19.5,20.5,21.5,22.5,23.5,24.5,
     $          25.5,26.5,27.5,28.5,29.5,30.5,
     $          31.5,32.5,33.5,34.5,35.5,36.5/
C
C  PWRKP VALUES COME FROM INTEGRATING HARDY MODEL OVER LATITUDE USING
C  PROGRAM SUMHDY (NOVEMBER, 1989)
C
      DATA PWRKP /.161,.171,.156,.168,.140,.127,.126,.109,.095,.097,
     1           .131,.222,.280,.209,.147,.142,.168,.211,.185,.159,
     1           .173,.189,.174,.156,.161,.171,
     2           .201,.146,.132,.164,.168,.161,.171,.226,.290,.303,
     2           .320,.387,.483,.520,.454,.424,.453,.464,.403,.353,
     2           .346,.326,.285,.243,.201,.146,
     3           .280,.183,.156,.207,.271,.258,.254,.317,.430,.532,
     3           .624,.751,.898,1.05,1.14,1.15,.959,.689,.615,.605,
     3           .564,.515,.455,.388,.280,.183,
     4           .263,.209,.224,.271,.300,.368,.585,.915,1.09,1.01,
     4           .973,1.23,1.64,1.83,1.72,1.61,1.63,1.60,1.31,.999,
     4           .872,.841,.688,.428,.263,.209,
     5           .263,.189,.166,.195,.215,.360,.699,1.01,1.48,1.65,
     5           1.57,1.57,1.92,2.18,1.84,1.55,1.76,2.25,2.12,1.49,
     5           1.21,1.10,.784,.433,.263,.189,
     6           .287,.229,.265,.340,.550,.880,.969,1.39,2.19,2.24,
     6           1.72,1.62,2.00,2.41,1.91,1.84,2.58,2.98,2.34,1.69,
     6           1.32,.978,.644,.425,.287,.229,
     7           .360,.315,.279,.309,.472,.849,1.60,2.42,3.12,3.27,
     7           3.15,3.99,6.21,7.26,5.63,4.19,4.45,5.02,4.12,2.89,
     7           2.06,1.38,.730,.431,.360,.315/
C
      PI=ATAN2(0.,-1.)
      DLTDJ=24./FLOAT(LTDIM-JWRAP)
      BKP=FKP+1.
      IF(BKP.GT.7.) BKP=7.
C
C  CHECK DIMENSION COMPATIBILITY
      IF(JDIM.NE.LTDIM) THEN
         WRITE(6,*) '**** DIMENSION ERROR IN SUBROUTINE PWRCAL ****'
         WRITE(6,*) '           JDIM=',JDIM,' LTDIM=',LTDIM
         WRITE(6,*) '**** STOPPING PROGRAM IN PWRCAL ****'
         STOP
      END IF
C
C  COMPUTE PTOT AND PWRTOT


      DO 10 J=JWRAP,LTDIM-1
         IMIN=NINT(BNDLOC(J,ITMCUR)+.5)
         FRACT(J)=0.5+.3*SIN(ALOCT(IMIN,J))
         TIME=ALOCT(IMIN,J)*12./PI+12.
```

```
C
          DO 20 N=1,25
             IF(TIME.GT.XMLT(N).AND.TIME.LE.XMLT(N+1)) THEN
                F=(TIME-XMLT(N))/(XMLT(N+1)-XMLT(N))
                BXMLT=FLOAT(N)+F
                GO TO 15
             END IF
   20     CONTINUE
             WRITE(6,*)'**** ERROR IN PWRCAL ****'
             WRITE(6,*) 'J=',J,' TIME=',TIME
             WRITE(6,*) 'STOPPING PGM IN PWRCAL'
             STOP
   15     CONTINUE
C
          PTOT(J)=G3NTRP(PWRKP,26,7,1,BXMLT,BKP,1.)
          PTOT(J)=PTOT(J)*DLTDJ
          PWRTOT(J)=PTOT(J)*(1.-FRACT(J))
C
          PWRMDL(J)=0.
C
          IMAX=LATDIM
C
       DO 30 I=IMIN,IMAX
          PWRMDL(J)=PWRMDL(J)+FLXSUM(I,J,1)*ALPHA(I)*BETA(I)
   30  CONTINUE
          PWRMDL(J)=1.E-6*DLAM*DPSI*RI**2*PWRMDL(J)
   40  CONTINUE
C
C PWRMDL(J) IS IN UNITS OF GW/GRID LINE


          RENORM(J)=PWRMDL(J)/PWRTOT(J)
   10  CONTINUE
C  CIRCULARIZE
          RENORM(1)=RENORM(JDIM-2)
          RENORM(2)=RENORM(JDIM-1)
          RENORM(JDIM)=RENORM(JWRAP)
          PTOT(1)=PTOT(JDIM-2)
          PTOT(2)=PTOT(JDIM-1)
          PTOT(JDIM)=PTOT(JWRAP)

          FRACT(1)=FRACT(JDIM-2)
          FRACT(2)=FRACT(JDIM-1)
          FRACT(JDIM)=FRACT(JWRAP)
C
          PWRTOT(1)=PWRTOT(JDIM-2)
          PWRTOT(2)=PWRTOT(JDIM-1)
          PWRTOT(JDIM)=PWRTOT(JWRAP)
C
          PWRMDL(1)=PWRMDL(JDIM-2)
          PWRMDL(2)=PWRMDL(JDIM-1)
          PWRMDL(JDIM)=PWRMDL(JWRAP)
C
C DEBUG 6.06.90 $$$$
      CALL OUTP(PWRMDL,1,JDIM,1,1,1,1,JDIM,1,0.0,'PWRMDL',6,132)
      CALL OUTP(PTOT,1,JDIM,1,1,1,1,JDIM,1,0.0,'PTOT',6,132)
      CALL OUTP(PWRTOT,1,JDIM,1,1,1,1,JDIM,1,0.0,'PWRTOT',6,132)
      CALL OUTP(RENORM,1,JDIM,1,1,1,1,JDIM,1,0.0,'RENORM',6,132)
C  END DEBUG 6.06.90 $$$$
C
C     WRITE(6,*) 'A(1)= ',A(1),'  B(1)= ',B(1),'  DX(1)= ',DX(1)
C     WRITE(6,*) 'DY(1)= ',DY(1)
C
      DO 100 J=1,JDIM
```

```
         IMIN=NINT(BNDLOC(J,ITMCUR)+.5)
         PT=FRACT(J)*PTOT(J)
         COLATA=THET(A(1),B(1),DX(1),DY(1),ALOCT(IMIN,J))*PI/180.
C        WRITE(6,*) 'J= ',J,'  COLATA= ',COLATA,'   PT= ',PT
C
C  FIND I VALUE SUCH THAT COLAT(I,J) .LT. COLATA .LT. COLAT(I+1,J)
C
         DO 110 II=1,IMIN-1
            I=II
            IF(COLATA.LT.COLAT(I+1,J).AND.
     2        COLATA.GE.COLAT(I,J)) GO TO 120
 110  CONTINUE
C
C  IF PROGRAM REACHES THIS PT, WE'VE GOT PROBLEMS
C
         WRITE(6,*) 'PROBLEMS IN PWRCAL ROUTINE; COLATA= ',COLATA,
     2          '   IMIN= ',IMIN
         WRITE(6,*) 'J= ',J
         WRITE(6,*) 'STOPPING PROGRAM IN PWRCAL'
         STOP
C
C
 120  CONTINUE
C
         BIL=FLOAT(I)+(COLATA-COLAT(I,J))/(COLAT(I+1,J)-COLAT(I,J))
C
C  DETERMINE CONSTANTS AX,BX, AND CX
C
C
C
C
C
         P1=1.E-6*FLXSUM(IMIN+1,J,1)*ALPHA(IMIN+1)*BETA(IMIN+1)*
     2            DLAM*DPSI*RI**2
         P2=1.E-6*FLXSUM(IMIN+2,J,1)*ALPHA(IMIN+2)*BETA(IMIN+2)*
     2            DLAM*DPSI*RI**2
         FIZ=FLOAT(IMIN)-BIL
         FI1=FLOAT(IMIN+1)-BIL
         FI2=FLOAT(IMIN+2)-BIL
         AX=(PT-((P1*FI2**3-P2*FI1**3)/3.+(P2*FI1**2-P1*FI2**2)*FIZ/4.)
     2         *FIZ**3/(FI1**2*FI2**2))/((FIZ**2/2.)*(1.-2.*(FI1+FI2)*
     3       FIZ/(3.*FI1*FI2)+FIZ**2/(2.*FI1*FI2)))
C
C
         BX=(P1*FI2**3-P2*FI1**3-AX*FI1*FI2*(FI1+FI2))/
     2         (AX*FI1**2*FI2**2)
C
         CX=(P2*FI1**2-P1*FI2**2+AX*FI1*FI2)/(AX*FI1**2*FI2**2)
C
         ILIM=BIL+.5
C     IF(J.EQ.12) THEN
C        WRITE(6,*) 'I,IMIN,COLATA,BIL,P1,P2,FIZ,FI1,FI2,AX,BX,CX'
C        WRITE(6,*) I,IMIN,COLATA,BIL,P1,P2,FIZ,FI1,FI2,AX,BX,CX
C     END IF
C
         DO 130 I=1,IMIN
            FIM=FLOAT(I)-.5-BIL
            FIP=FLOAT(I)+.5-BIL
            IF(I.LT.ILIM) THEN
               PS=0.
            ELSE IF(I.EQ.ILIM) THEN
               PS=AX*(FIP**2/2.+BX*FIP**3/3.+CX*FIP**4/4.)
            ELSE
               PS=AX*((FIP**2-FIM**2)/2.
```

```
     2               +BX*(FIP**3-FIM**3)/3.
     3               +CX*(FIP**4-FIM**4)/4.)
            END IF
C
            IF(PS.GE.0.) THEN
               FLXSUM(I,J,1)=1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)
               EAVG(I,J,1)=(1.7-1.3*COS(ALOCT(I,J)))*1.E3
            ELSE
               GO TO 150
            END IF
C
  130 CONTINUE
C
         GO TO 200
C
  150 CONTINUE
C
C  TO GET HERE MEANS PS .LT. 0 FOR SOME GRID PT
         ILIM=BIL-.5
         DD=FLOAT(IMIN-ILIM)+.5
         PS=PT/DD
         DO 160 I=ILIM,IMIN-1
            FLXSUM(I,J,1)=1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)
            EAVG(I,J,1)=(1.7-1.3*COS(ALOCT(I,J)))*1.E3
  160    CONTINUE
         FLXSUM(IMIN,J,1)=.5*1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)
         EAVG(IMIN,J,1)=3.
C
  200    CONTINUE
C
C  DEBUG PRINTOUT PARAMETERS
C
         BI1=FLOAT(IMIN+1)-BIL
         BI2=FLOAT(IMIN+2)-BIL
C
         PX1=AX*(BI1+BX*BI1**2+CX*BI1**3)
         PX2=AX*(BI2+BX*BI2**2+CX*BI2**3)
C
         IL=BIL-.5
         IH=IMIN-1
         PXT=0.
         DO 210 I=IL,IH
           PXT=PXT+1.E-6*FLXSUM(I,J,1)*ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2
  210    CONTINUE
         BIX=FLOAT(IMIN)-BIL
         BIY=FLOAT(IMIN)-.5-BIL
         PXT=PXT+AX*((BIX**2-BIY**2)/2.+BX*(BIX**3-BIY**3)/3.
     2              +CX*(BIX**4-BIY**4)/4.)
C
C
C
  100 CONTINUE
C
C
C
      RETURN
      END
      FUNCTION WKRATE(RMID,RPPMID,PMID,ENRG)
C
C  PURPOSE:   FUNCTION SUBPROGRAM TO EVALUATE WEAK PRECIPITATION LOSS
C             RATE AS DESCRIBED IN R.A. WOLF DOCUMENT 'PITCH ANGLE
C             SCATTERING ASSUMPTIONS FOR 1/90 MSM RUNS' DATED 1/4/90.
C
C  PROGRAMMER: R. W. SPIRO              DATE: 01.05.90
```

```
C
C  INPUT PARAMETERS:
C        RMID            R VALUE AT PT WHERE WKRATE IS TO BE EVALUATED
C                            (RE)
C        RPPMID          ESTIMATE OF PLASMAPAUSE RADIUS (RE)
C        PMID            LOCAL TIME HOUR ANGLE IN EQUATORIAL PLANE
C                            (RADIANS FROM LOCAL NOON)
C        ENRG            PARTICLE ENERGY (EV)
C
      IF(RMID.LE.RPPMID) THEN
C  USE LYONS RESULTS
         WKRATE=RLYONS(RMID,ENRG)
         GO TO 100
      ELSE IF(ENRG.LE.20.E3) THEN
         WKRATE=0.
         GO TO 100
      ELSE
         WKGEO=MIN(ENRG/20.E3-1.,1.)*(1.+(9./11.)*COS(PMID))/1200.
C
         IF(RMID.GT.6.6) THEN
            WKRATE=WKGEO
            GO TO 100
         ELSE
            XLYONS=RLYONS(RPPMID,ENRG)
            WKRATE=((6.6-RMID)/(6.6-RPPMID))*XLYONS+
     2              ((RMID-RPPMID)/(6.6-RPPMID))*WKGEO
         END IF
      END IF
C
  100 CONTINUE
      IF(WKRATE.GT.10.) THEN
         WRITE(6,800) RMID,RPPMID,PMID,ENRG,WKRATE
  800 FORMAT(1X,'WKRATE.GT.10'/1X,'RMID=',F6.3,2X,'RPPMID=',F6.3,2X,
     2          'PMID=',F6.3,2X,'ENRG=',1PE12.3,2X,'WKRATE=',E12.3)
      END IF
      RETURN
      END
      REAL FUNCTION RLYONS(RMID,ENRG)
C
C  PURPOSE:  FUNCTION SUBPROGRAM TO EVALUATE LYONS-BASED WEAK LOSS RATE.
C
C  PROGRAMMER:  R.W. SPIRO             DATE: 01.05.90
C
      IF(ENRG.LE.50.E3) THEN
         XLOGLO=(4.9365+15.8621)-6.06897*RMID+0.551724*RMID**2
         XLOGHI=(4.9365+13.5172)-5.58621*RMID+0.551724*RMID**2
         F=(ENRG-20.E3)/(50.E3-20.E3)
      ELSE IF(ENRG.GT.50.E3.AND.ENRG.LE.200.E3) THEN
         XLOGLO=(4.9365+13.5172)-5.58621*RMID+0.551724*RMID**2
         XLOGHI=(4.9365+1.15517)-0.42069*RMID+0.0275862*RMID**2
         F=(ENRG-50.E3)/(200.E3-50.E3)
      ELSE
         XLOGLO=(4.9365+1.15517)-0.42069*RMID+0.0275862*RMID**2
         XLOGHI=(4.9365+3.89655)-1.75862*RMID+0.206897*RMID**2
         F=(ENRG-200.E3)/(500.E3-200.E3)
      END IF
C
      XLOG=(1.-F)*XLOGLO+F*XLOGHI
C
      TAU=10.**XLOG
C
      XLYONS=1./TAU
      RLYONS=XLYONS
C
```

```
C
      RETURN
      END
      FUNCTION G3TRPA(A,IMAX,JMAX,KMAX,BI,BJ,BK)
C
C  VERSION 1.00                             DATE: 01.09.90

C
C  PURPOSE: FUNCTION SUBPROGRAM TO PERFORM A GENERAL 3-D LINEAR
C    INTERPOLATION OF ANGULAR ARRAY A(I,J,K) AT PT(BV(1),BV(2),BV(3))
C
C  INPUT:
C       A                         3-D ARRAY TO BE INTERPOLATED
C                                 INTERPOLATE A
C
C
C
        COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION A(IMAX,JMAX,KMAX)
      DIMENSION NDX(3),NDIM(3),BV(3),COEF(3,2)
C
C  PREPARE INDICES FOR INTERPOLATION
C
      PI=ATAN2(0.,-1.)
      NDIM(1)=IMAX
      NDIM(2)=JMAX
      NDIM(3)=KMAX
      BV(1)=BI
      BV(2)=BJ
      BV(3)=BK
      DO 10 L=1,3
         NDX(L)=BV(L)
         IF(NDX(L).LT.1) NDX(L)=1
         IF(NDX(L).GT.NDIM(L)-1) NDX(L)=NDIM(L)-1
         IF(NDX(L).LE.0) NDX(L)=1
C
         FNDX=REAL(NDX(L))
         COEF(L,1)=1.-BV(L)+FNDX
         COEF(L,2)=BV(L)-FNDX
   10 CONTINUE
C
C
      G3TRPA=0.
      IC=0
      KSTOP = MIN(KMAX,2)
      JSTOP = MIN(JMAX,2)
      DO 20 I=1,2
      DO 20 J=1,JSTOP
      DO 20 K=1,KSTOP
         IC=IC+1
         AVAL=A(NDX(1)+I-1,NDX(2)+J-1,NDX(3)+K-1)
         IF(IC.EQ.1) THEN
C  USE FIRST VALUE OF ARRAY TO SET MODULUS
            REFMOD=AVAL
         END IF
C
         IF((AVAL-REFMOD).GT.PI) THEN
            AVAL=AVAL-2.*PI
         ELSE IF((AVAL-REFMOD).LE.-PI) THEN
            AVAL=AVAL+2.*PI
         END IF
```

```
C                                                                       MSM72320
         G3TRPA=G3TRPA+                                                 MSM72330
     1          COEF(1,I)*COEF(2,J)*COEF(3,K)*AVAL                      MSM72340
   20 CONTINUE                                                          MSM72350
C                                                                       MSM72360
C  GET RESULT IN A REASONABLE MODULUS                                   MSM72370
      IF(G3TRPA.GT.2.*PI) THEN                                          MSM72380
         G3TRPA=G3TRPA-2.*PI                                            MSM72390
      ELSE IF(G3TRPA.LT.0.) THEN                                        MSM72400
         G3TRPA=G3TRPA+2.*PI                                            MSM72410
      END IF                                                            MSM72420
      RETURN                                                            MSM72430
      END                                                               MSM72440
      SUBROUTINE THRCAL(FKP,FNRGLG,NRGNDX,KPNDX,IRDIM,NRGDIM,KPDIM,     MSM72450
     2                  KPPLUS,ISPDIM,FLXMAT,THR)                       MSM72460
C                                                                       MSM72470
C  PURPOSE:  SUBROUTINE TO CALCULATE THRESHOLD FLUX AT R=3,4,6.6, AND   MSM72480
C          13 RE FOR KP=FKP AND LOG(ENERGY)=FNRGLG.  THR IS IN UNITS    MSM72490
C          OF PARTICLES/CM**2/S/SR/EV.                                  MSM72500
C                                                                       MSM72510
C  PROGRAMMER: R.W. SPIRO              DATE: 05.09.90                   MSM72520
C                                                                       MSM72530
C                                                                       MSM72540
      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM), THR(4),RTHR(4)      MSM72550
      DIMENSION EZERO(4,7), RKAP(4)                                     MSM72560
C                                                                       MSM72570
      DATA RTHR /3.,4.,6.6,13./                                         MSM72580
C                                                                       MSM72590
      DATA EZERO /23.5E3,16.E3,21.E3,.49275E3,                          MSM72600
     1            23.5E3,16.E3,4.8324E3,.625E3,                         MSM72610
     2            23.5E3,16.E3,4.814E3,.75725E3,                        MSM72620
     3            23.5E3,16.E3,4.7978E3,.8895E3,                        MSM72630
     4            23.5E3,16.E3,4.816E3,1.02175,                         MSM72640
     5            23.5E3,16.E3,4.8943E3,1.154E3,                        MSM72650
     6            23.5E3,16.E3,5.0752E3,1.28625E3/                      MSM72660
C                                                                       MSM72670
      DATA RKAP /2.8,2.0,6.,6./                                         MSM72680
C                                                                       MSM72690
C                                                                       MSM72700
      ETOP=40.E3                                                        MSM72710
      ELOW=20.E3                                                        MSM72720
      ENRG=10.**FNRGLG                                                  MSM72730
C                                                                       MSM72740
      IF(ENRG.LE.ELOW) THEN                                             MSM72750
         THR(1)=FLXMAT(1,NRGNDX,1,1)                                    MSM72760
         THR(2)=FLXMAT(2,NRGNDX,1,1)                                    MSM72770
         THR(3)=FLXMAT(3,NRGNDX,1,1)                                    MSM72780
         THR(4)=FLXMAT(4,NRGNDX,1,1)                                    MSM72790
C                                                                       MSM72800
      ELSE IF(ENRG.GE.ETOP) THEN                                        MSM72810
         DO 10 N=1,4                                                    MSM72820
            RK=RKAP(N)                                                  MSM72830
            EZ=EZERO(N,KPNDX)                                           MSM72840
C                                                                       MSM72850
            IF(RK.GE.6.0)THEN                                           MSM72860
             ER=(1./EZ)*((RK-1.)/RK)*(ENRG/EZ)*(1.+ETOP/(RK*EZ))**RK/   MSM72870
     2          (1.+MIN(ENRG/RK/EZ,1.E5))**(RK+1.)/(1.+ETOP/EZ)         MSM72880
            ELSE                                                        MSM72890
             ER=(1./EZ)*((RK-1.)/RK)*(ENRG/EZ)*(1.+ETOP/(RK*EZ))**RK/   MSM72900
     2           (1.+ENRG/RK/EZ)**(RK+1.)/(1.+ETOP/EZ)                  MSM72910
            ENDIF                                                       MSM72920
C                                                                       MSM72930
            THR(N)=ALOG10(5.E7*ER*(6.6/RTHR(N))**4)                     MSM72940
C                                                                       MSM72950
```

```
   10    CONTINUE
C
      ELSE
C  ENRG IS BETWEEN ELOW AND ETOP
C
         DO 20 N=1,4
            RK=RKAP(N)
            EZ=EZERO(N,KPNDX)
C
            ERTOP=(1./EZ)*((RK-1.)/RK)*(ETOP/EZ)/
     2              (1.+ETOP/RK/EZ)/(1.+ETOP/EZ)
C
            THRTOP=ALOG10(5.E7*ERTOP*(6.6/RTHR(N))**4)
C
            THRLOW=FLXMAT(N,NRGNDX,1,1)
C
            THR(N)=((ETOP-ENRG)*THRLOW+(ENRG-ELOW)*THRTOP)/(ETOP-ELOW)
C
   20    CONTINUE
C
      END IF
C
      RETURN
      END
       SUBROUTINE FLXDFL(ITM,FKP,R,P,ALAM,VM,IRDIM,NRGDIM,KPDIM,KPPLUS,
     1   ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,ALMDEL,LATDIM,
     2   LTDIM,IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,IEBEG,IEEND)
C
C  SUBROUTINE WHICH USES THE FLXMAT ARRAYS TO
C   CALCULATE THE ENERGY-DEPENDENT ETA ARRAYS.  THIS IS THE
C   DEFAULT MODEL WHEN ONLY KP IS AVAILABLE TO RUN THE PROGRAM.
C
      PARAMETER(JJDIM=51)
      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM),
     1   ALAM(IEDIM),VM(LATDIM,LTDIM,ITMDIM),
     2   FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM),
     3   FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM),
     4   BNDLOC(LTDIM,ITMDIM),EFLUX(LATDIM,LTDIM,IEDIM),IBLOC(JJDIM),
     5   ETA(LATDIM,LTDIM,IEDIM),ALMDEL(IEDIM),IFLAV(IEDIM)
C
      IF(JJDIM.NE.LTDIM)THEN
       WRITE(6,*)' DIMENSION ERROR IN FLXDFL SUBROUTINE, PROGRAM STOPS.'
       STOP
      ENDIF
C
      DO 10 J=1,LTDIM
      IBLOC(J)=IFIX(BNDLOC(J,ITM))
      DO 20 I=IBLOC(J),LATDIM
       DO 30 K=IEBEG,IEEND
        EFLUX(I,J,K)=0.0
30     CONTINUE
20    CONTINUE
10    CONTINUE
      DO 50 J=1,LTDIM
       DO 60 I=IBLOC(J),LATDIM
        DO 70 K=IEBEG,IEEND
         ISP=IFLAV(K)
         ENRG=ABS(ALAM(K))*VM(I,J,ITM)
          FLXOUT=FLXVAL(ISP,FKP,R(I,J,ITM),P(I,J,ITM),ENRG,IRDIM,
     1      NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP)
         IF(IFLAV(K).EQ.1)THEN
          XMSCNT=7.392E-16
C         IF((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13))THEN
C         WRITE(6,*)' I,J,K,FLUX,ENRG ',I,J,K,FLXOUT,ENRG
```

```
C         WRITE(6,*)' VM,ALMDEL,MSCNST',VM(I,J,ITM),ALMDEL(K),MSCNST
C         WRITE(6,*)' IFLAV ',IFLAV(K)
C         ENDIF
          ETA(I,J,K)=(FLXOUT*ALMDEL(K))/(XMSCNT*SQRT(ABS(ALAM(K)))
     1                *VM(I,J,ITM))
C         IF((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13))THEN
C         WRITE(6,*)' ETA ', ETA(I,J,K)
C         ENDIF
         ELSE
C           IONS
          XMSCNT=1.731E-17
          IF(IFLAV(K).EQ.2) THEN
           ATOMWT=1.0
          ELSE
           ATOMWT=16.0
          ENDIF
          ETA(I,J,K)=(FLXOUT*ALMDEL(K)*SQRT(ATOMWT))/(XMSCNT*
     1       SQRT(ABS(ALAM(K)))*VM(I,J,ITM))
         ENDIF
70      CONTINUE
60     CONTINUE
50    CONTINUE
C     DO 100 K=IEBEG,IEEND
C      IF((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13))THEN
C       WRITE(18,*)' K = ',K
C       WRITE(6,*) ' K = ',K
C       CALL OUTP(ETA(1,1,K),LATDIM,LTDIM,10,50,1,3,LTDIM-1,1,
C    1    0.,'ETA',6,132)
C      ENDIF
100   CONTINUE
      RETURN
      END
      SUBROUTINE RMVBSH(WORK,LATDIM,LTDIM)
C
C  TEMPORARY FIX TO THE B-FIELD MATRICES TO REMOVE THE LARGE DATA
C   VALUES SIGNIFYING THAT THE FIELD LINES HAD 'GONE INTO THE BUSHES'.
C   IT LOOKS FOR THE CLOSEST VALUE IN J AND LINEARLY INTERPOLATES FOR
C   THE I VALUES.
C
      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
      BFBUSH = 1.E20
      DO 10 N1=1,2
       DO 20 N2=1,2
        DO 30 N3=1,2
         DO 40 N4=1,2
          DO 50 N5=1,2
           DO 60 J=1,LTDIM
            DO 70 I=1,LATDIM
             IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH)GO TO 70
             IDO=I
999          CONTINUE
             IDO=IDO+1
             IF(WORK(IDO,J,N1,N2,N3,N4,N5).GT.BFBUSH)GO TO 999
             COEF1=WORK(IDO,J,N1,N2,N3,N4,N5)
             COEF2= WORK(IDO+1,J,N1,N2,N3,N4,N5)
             STRTLN=COEF2-COEF1
             DO 80 ICHK=IDO-1,1,-1
              WORK(ICHK,J,N1,N2,N3,N4,N5)=WORK(ICHK+1,J,N1,N2,N3,N4,N5)
     1             - STRTLN
80           CONTINUE
70          CONTINUE
60         CONTINUE
50        CONTINUE
40       CONTINUE
```

```
30        CONTINUE
20       CONTINUE
10      CONTINUE
        RETURN
        END
        SUBROUTINE DSTDFL(MODE,AUGPAR,NAUGEL,ITMDIM,ITM,ITMMAX,TIMTAG)
C
C     SUBROUTINE TO CALCULATE THE DEFAULT DST VALUES IF NO
C     VALUE IS FOUND IN THE ENVIRONMENTAL DATA BASE. (MODE IS FALSE)
C
        INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,DEQDT,IPATT
C
        COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,IPATT,DDST,DEQDT
C
        DIMENSION AUGPAR(NAUGEL,ITMDIM),TIMTAG(ITMDIM)
        DIMENSION YA(2),XA(2)
C
        LOGICAL*1 MODE
        IF(MODE)RETURN
        IF(AUGPAR(KP,ITM).LT.4.0)THEN
         AUGPAR(DST,ITM) = -AUGPAR(KP,ITM)*4.0
        ELSE
         AUGPAR(DST,ITM) = -20.0 -(((AUGPAR(KP,ITM) * 10.0) - 40.0) * 2.6)
        ENDIF
        IF(ITM.GT.1)THEN
         YA(1)=AUGPAR(DST,ITM-1)
         YA(2)=AUGPAR(DST,ITM)
         XA(1)=TIMTAG(ITM-1)
         XA(2)=TIMTAG(ITM)
         CALL DTNTRP(TIMTAG(ITM),2,XA,YA,Y,DYDX,DELTA,IERR)
         AUGPAR(DDST,ITM-1)=DYDX*3600.
        ENDIF
        IF(ITM.EQ.ITMMAX)AUGPAR(DDST,ITM)=AUGPAR(DDST,ITM-1)
        RETURN
        END
        SUBROUTINE EQTDFL(MODE,AUGPAR,NAUGEL,ITMDIM,ITM,ITMMAX,TIMTAG)
C
C    SUBROUTINE TO CALCULATE  A KP-BASED VALUE FOR THE
C    EQUATORWARD EDGE OF THE AURORAL OVAL BASED ON THE WORK OF
C    GUSSENHOVEN ET. AL., 1983.
C   IF MODE IS TRUE A VALUE HAS ALREADY BEEN RECEIVED FROM THE
C    ENVIRONMENTAL DATA BASE AND THE SUBROUTINE JUST RETURNS.
C
        INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,DEQDT,IPATT
C
        COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
     1   EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
     2   GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
     3   STAND,PCP,IPATT,DDST,DEQDT
C
        DIMENSION AUGPAR(NAUGEL,ITMDIM),TIMTAG(ITMDIM)
        DIMENSION XA(2),YA(2)
C
        LOGICAL MODE*1
        IF(MODE)RETURN
```

```
      AUGPAR(EQEDGE,ITM) = 67.78 - (2.068 * AUGPAR(KP,ITM))
      IF(ITM.GT.1)THEN
       YA(1)=AUGPAR(EQEDGE,ITM-1)
       YA(2)=AUGPAR(EQEDGE,ITM)
       XA(1)=TIMTAG(ITM-1)
       XA(2)=TIMTAG(ITM)
       CALL DTNTRP(TIMTAG(ITM),2,XA,YA,Y,DYDX,DELTA,IERR)
       AUGPAR(DEQDT,ITM-1)=DYDX*3600.
      ENDIF
      IF(ITM.EQ.ITMMAX)AUGPAR(DEQDT,ITM)=AUGPAR(DEQDT,ITM-1)
      RETURN
      END
      SUBROUTINE PCPDFL(MODE,XKP, PCP)
C
C  THIS SUBROUTINE CALCULATES THE CROSS-POLAR CAP POTENTIAL (IN KV)
C   BASED ON THE WORK OF REIFF ET AL IF MODE IS FALSE.  IF MODE IS TRUE
C   THE POTENTIAL HAS ALREADY BEEN RECEIVED FROM THE ENVIRONMENTAL DATA
C   BASE AND THE SUBROUTINE JUST RETURNS.
C
      LOGICAL*1 MODE
      IF(MODE)RETURN
      PCP = 14.587 + (1.7 * (XKP * 10.0))
      RETURN
      END
      SUBROUTINE PATDFL(MODE,MODEBY,MODEBZ,BY,BZ,IPATT)
C
C  THIS SUBROUTINE USES THE BY AND BZ COMPONENTS OF THE IMF TO
C   DETERMINE THE ELECTRIC FIELD PATTERN TYPE TO BE USED.
C  THE DIFFERENT TYPES ARE:
C   BZ>0 - TYPE 0.0
C   BZ<0,BY>0 - TYPE 1.0
C   BZ<0,BY=0 - TYPE 2.0  (DEFAULT)
C   BZ<0,BY<0 - TYPE 3.0
C  IF MODE IS TRUE, THE PATTERN TYPE HAS ALREADY BEEN RECEIVED FROM
C   THE ENVIRONMENTAL DATA BASE AND THE SUBROUTINE JUST RETURNS.
C
      LOGICAL*1 MODE, MODEBY,MODEBZ
      IF(MODE)RETURN
      IF(.NOT.(MODEBZ))THEN
       XIPATT=2.0
      ELSE
       IF(BZ.GT.0.0)THEN
        XIPATT=0.0
       ELSE
        IF(.NOT.(MODEBY))THEN
         XIPATT=2.0
        ELSE
         IF(BY.GT.0.0)XIPATT=1.0
         IF(BY.EQ.0.0)XIPATT=2.0
         IF(BY.LT.0.0)XIPATT=3.0
        ENDIF
       ENDIF
      ENDIF
      RETURN
      END
      SUBROUTINE CLPDFL(MODE,CLAPSE)
C
C  CLPASE IS THE INDICATOR FOR WHETHER OR NOT TO USE THE COLLAPSED
C   TAIL VERSION OF THE MAGNETIC FIELD MODEL.  IF MODE IS FALSE, NO
C   VALUE HAS BEEN RETURNED FROM THE ENVIRONMENTAL DATA BASE.  THE
C   DEFAULT VALUE IS NO COLLAPSE (CLAPSE=1.0).
C
      LOGICAL*1 MODE
      IF(MODE)RETURN
```

```
      CLAPSE=1.0
      RETURN
      END
      SUBROUTINE STNDFL(MODE,MODVEL,MODDEN,SWVEL,SWDEN,XKP,STAND)
C
C  THIS SUBROUTINE CALCULATES THE STANDOFF DISTANCE.  IF SOLAR WIND
C    VELOCITY AND DENSITY DATA IS AVAILABLE SUBROUTINE STNDOF DOES THE
C    THE ACTUAL CALCULATION.  OTHERWISE THE DEFAULT IS A KP-BASED
C    FORMULA.
C
      LOGICAL*1 MODE,MODVEL,MODDEN
      IF(MODE)RETURN
      IF(MODVEL.AND.MODDEN)THEN
       CALL STNDOF(SWVEL,SWDEN,STAND)
      ELSE
       IF(XKP.LT.6.7)STAND=11.7-XKP
       IF(XKP.GE.6.7)STAND=5.0
      ENDIF
      RETURN
      END
      SUBROUTINE TILT(DAY,YEAR,XTILT)
C
C  SUBROUTINE TO CALCULATE THE EARTH'S TILT ANGLE.  CURRENTLY SET
C   TO 0.0
C
      XTILT=0.0
      RETURN
      END
C-----------------------------------------------------------------------

C                 ======
        SUBROUTINE AURL2S(FKP,COLAT,ALOCT,LATDIM,LTDIM,
     *                    AEFLUX,AEMEAN)
C                 ======
C       [Inputs]
C       FKP : The most recent Kp value
C       COLAT:An array listing pertinent colatitudes <40 deg
C       ALOCT:An array listing pertinent magnetic loca time
C       [Outputs]
C       AEFLUX:The integral ION energy flux (erg/cm2/sec)
C       AEMEAN:AVERAGE ENERGY IN EV
C       AFLUX :The integral ION number flux (/cm2/sec)
C       This subroutine calls REGEN() and EFUN()
C       June 04, 1989   by Akira NAGAI
C
C       Note: The feature is optimized for the speed.
C             At least 10KB static memory area is reqired
C             for REAL*2 in MSM Model.
C
        PARAMETER (IIDIM=62,JJDIM=51)
C
        COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1     LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2     LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3     LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C

        DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM),
     1            AFLUX(IIDIM,JJDIM),AEFLUX(LATDIM,LTDIM),
     2            AEMEAN(LATDIM,LTDIM),
     3  X1(JJDIM,0:6,2),X2(JJDIM,0:6,2),X3(JJDIM,0:6,2),
     4  X4(JJDIM,0:6,2),X5(JJDIM,0:6,2),X6(JJDIM,0:6,2),
     5  C1(0:12),C2(0:12),C3(0:12),C4(0:12),C5(0:12),C6(0:12)
```

```
      IKP=NINT(FKP)                                                    MSM76160
      IF((IPASS.EQ.0).AND.(IKP.EQ.0)) GOTO 60                          MSM76170
      IF(IKP.EQ.KP) GOTO 80                                            MSM76180
   60 KP=IKP                                                           MSM76190
      IF(KP.GT.6) KP=6                                                 MSM76200
                                                                       MSM76210
C     Just once is OK for this loop(10) for the entire run.            MSM76220
      DO 10 ICASE=1,2                                                  MSM76230
                                                                       MSM76240
        IF(IPASS.EQ.1) GOTO 80                                         MSM76250
                                                                       MSM76260
        IF(ICASE.EQ.1)THEN                                             MSM76270
         OPEN(UNIT=LIONNO,ACCESS='SEQUENTIAL',FILE='IONNUM',           MSM76275
     1        STATUS='OLD')                                            MSM76280
        ELSE                                                           MSM76290
         OPEN(UNIT=LIONEG,ACCESS='SEQUENTIAL',FILE='IONENG',           MSM76295
     1        STATUS='OLD')                                            MSM76300
        ENDIF                                                          MSM76310
C                                                                      MSM76320
        DO 20 IKP=0,6                                                  MSM76330
          DO 30 J=0,12                                                 MSM76340
           IF(ICASE.EQ.1)THEN                                          MSM76350
            READ(LIONNO,*) IDX,C1(J),C2(J),C3(J),C4(J),C5(J),C6(J)     MSM76360
           ELSE                                                        MSM76370
            READ(LIONEG,*) IDX,C1(J),C2(J),C3(J),C4(J),C5(J),C6(J)     MSM76380
           ENDIF                                                       MSM76390
   30     CONTINUE                                                     MSM76400
          CALL REGEN(LATDIM,LTDIM,ALOCT,C1,X1,JJDIM,IKP,ICASE)         MSM76410
          CALL REGEN(LATDIM,LTDIM,ALOCT,C2,X2,JJDIM,IKP,ICASE)         MSM76420
          CALL REGEN(LATDIM,LTDIM,ALOCT,C3,X3,JJDIM,IKP,ICASE)         MSM76430
          CALL REGEN(LATDIM,LTDIM,ALOCT,C4,X4,JJDIM,IKP,ICASE)         MSM76440
          CALL REGEN(LATDIM,LTDIM,ALOCT,C5,X5,JJDIM,IKP,ICASE)         MSM76450
          CALL REGEN(LATDIM,LTDIM,ALOCT,C6,X6,JJDIM,IKP,ICASE)         MSM76460
   20   CONTINUE                                                       MSM76470
        CLOSE(LIONNO)                                                  MSM76480
        CLOSE(LIONEG)                                                  MSM76490
   10 CONTINUE                                                         MSM76500
                                                                       MSM76510
      IPASS=1                                                          MSM76520
                                                                       MSM76530
   80 DO 40 IMLT=1,LTDIM                                               MSM76540
        DO 50 ILAT=1,LATDIM                                            MSM76550
          H=90.-ABS(COLAT(ILAT,IMLT))*180/3.14159                      MSM76560
          FX=EFUN(1,KP,H,X1(IMLT,KP,1),X2(IMLT,KP,1),                  MSM76570
     *    X3(IMLT,KP,1),X4(IMLT,KP,1),X5(IMLT,KP,1),X6(IMLT,KP,1))     MSM76580
C         Unit from 1/cm2/sec/str to 1/cm2/sec                         MSM76590
          AFLUX(ILAT,IMLT) = (10.**FX)*3.1415265358                    MSM76600
          FX=EFUN(2,KP,H,X1(IMLT,KP,2),X2(IMLT,KP,2),                  MSM76610
     *    X3(IMLT,KP,2),X4(IMLT,KP,2),X5(IMLT,KP,2),X6(IMLT,KP,2))     MSM76620
C         Unit from keV/cm2/sec/str to erg/cm2/sec                     MSM76630
          AEFLUX(ILAT,IMLT)= (10.**FX)*3.14159265358*1.6021E-9         MSM76640
          IF(AFLUX(ILAT,ILMT).NE.0.0)AEMEAN(ILAT,IMLT) =               MSM76650
     1      AEFLUX(ILAT,IMLT)/AFLUX(ILAT,IMLT)                         MSM76660
          AEMEAN(ILAT,IMLT)=AEMEAN(ILAT,IMLT)*6.24E11                  MSM76670
   50   CONTINUE                                                       MSM76680
   40 CONTINUE                                                         MSM76690
                                                                       MSM76700
   99 RETURN                                                           MSM76710
      END                                                              MSM76720
C-------------------------------------------------------------------   MSM76730
C              ====                                                    MSM76740
      FUNCTION EFUN(ICASE,KP,H,R0,H0,H1,S0,R1,S2)                      MSM76750
C              ====                                                    MSM76760
C    This function is called only by AURL2S() and this does not        MSM76770
```

```
                XLO=(ENERGY-HAFCHN(NCHNL))/(HAFCHN(NCHNL-1)-
     2                                    HAFCHN(NCHNL))
C
   48     CONTINUE
             XL1=1.-XLO
             XSTFLX=XLO*SATFLX(KK)+XL1*SATFLX(KK+1)
C
            DO 50 INO=1,LATDIM
             DO 60 JNO=1,LTDIM
              FLXNTR(INO,JNO,1)=FLUX(INO,JNO,KNERGY)
              FLXNTR(INO,JNO,2)=FLUX1(INO,JNO,KNERGY)
60           CONTINUE
50          CONTINUE
            FLUXT=G3NTRP(FLXNTR,LATDIM,LTDIM,2,BI,BJ,TMTRP1)
C
C
             FLUXT=FLUXT*1000.0
           IF(FLUXT.LE.0.0)THEN
            WRITE(6,*)' BAD FLUX VALUE CALCULATED IN SUBROUTINE OPTERR'
            WRITE(6,*)' UNABLE TO CONTINUE'
            WRITE(6,*)' VM, FLUX = ',CALVM,FLUXT
            STOP
          ENDIF
C
      WRITE(6,*) 'K=',KNERGY,'  SAT FLUX=',XSTFLX,'  MODEL FLUX=',
     2           LOG10(FLUXT),'  ALAM=',ALAM(KNERGY)
C
           IF(STNDEV(KNERGY).EQ.-1.0)THEN
            STNDEV(KNERGY)=(LOG10(FLUXT)-XSTFLX)**2
           ELSE
            STNDEV(KNERGY)=STNDEV(KNERGY)
     1          +((LOG10(FLUXT)-XSTFLX)**2)
           ENDIF
C
           NPTS(KNERGY)=NPTS(KNERGY)+1
           COEF(KNERGY)=COEF(KNERGY)+(LOG10(FLUXT)-XSTFLX)
C          WRITE(6,*)' NOCHNL,STNDEV,COEF,KNERGY ',NOCHNL-1,
C    1         STNDEV(KNERGY),COEF(KNERGY),KNERGY,FLUXT,XSTFLX
          ENDIF
45      CONTINUE
         XCHNCK=XCHNL
         IF(ISP.EQ.1) THEN
           READ(99,*,END=400)XCHNL,YEAR,DECDAY,LAT,LONG,ALT,MLT
         ELSE
           READ(98,*,END=400) XCHNL,YEAR,DECDAY,LAT,LONG,ALT,MLT
         END IF
C
         IF(XCHNCK.NE.XCHNL)THEN
          WRITE(6,*)' NUMBER OF SATELLITE ENERGY CHANNELS CHANGED IN'
          WRITE(6,*)' MID-RUN.  UNABLE TO COMPUTE ERROR.  RETURNING TO'
          WRITE(6,*)' MAIN PROGRAM.'
          RETURN
         ENDIF
         TIMCHK=TCONV2(YEAR,DECDAY,YRCHCK,DYCHCK)
         WRITE(6,*)' NPTS,TIMCHK,TIMTAG,ITM ',NPTS,TIMCHK,TIMTAG(ITM+1),
     1      ITM
C
         IF(TIMCHK.LT.TIMTAG(ITM+1))GO TO 200
C
C  COMPUTE STANDARD DEVIATION AND TIME SHIFT FOR THIS TIME INTERVAL
C
  400 CONTINUE
C
       IF(ISP.EQ.NSP) THEN
```

```
         DO 70 KNERGY=1,IEMAX
          ENERGY=(ABS(ALAM(KNERGY))*VMGEO)/1000.00
          WRITE(6,*)' KNERGY,STNDEV ',KNERGY,STNDEV(KNERGY)
          IF(NPTS(KNERGY).NE.0)THEN
            STNDEV(KNERGY)=SQRT(STNDEV(KNERGY)/NPTS(KNERGY))
            WRITE(6,*)' STANDARD DEVIATION (LOG10) = ',STNDEV(KNERGY),
     1        ' FOR ENERGY ',ENERGY,' (KEV)'
           COEF(KNERGY)=COEF(KNERGY)/NPTS(KNERGY)
           ERSHFT(KNERGY)=COEF(KNERGY)
           WRITE(6,*)' PARALLEL SHIFT (LOG10) = ',COEF(KNERGY),
     1         ' FOR ENERGY ',ENERGY,' (KEV)'
          ELSE
            ERSHFT(KNERGY)=0.
          ENDIF
70       CONTINUE
C
C  WRITE STANDARD DEVIATION AND SHIFT TO OUTPUT DATA SET
         IRDOUT=IREC+1
        WRITE(6,*) 'WRITING INTO SHFT FILE   IRDOUT=',IRDOUT,
     2           'IEMAX=',IEMAX,' ERSHFT=',ERSHFT
         CALL WRTVEC(LUSHFT,IRDOUT,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
        END IF
C
30       CONTINUE
C
   15 CONTINUE
C
        RETURN
        END
        SUBROUTINE WRTVEC(NOUNIT,IRDOUT,IEMAX,IEDIM,ALAM,THRSH,
     1  ERSHFT,IFLAV)
C
C  SUBROUTINE TO WRITE
C    ALAMS,THRESHOLD ENERGIES, AND ERROR
C   FOR EACH OF THE ENERGY CHANNELS IN THIS RUN
C
C
        DIMENSION ALAM(IEDIM),THRSH(IEDIM),ERSHFT(IEDIM),IFLAV(IEDIM)
C
        IRECL=4*IEDIM+1
        OPEN(UNIT=NOUNIT,ACCESS='DIRECT',RECL=IRECL,STATUS='UNKNOWN')
C
C  WRITE THE INFORMATION
C
        WRITE(NOUNIT,REC=IRDOUT)IEMAX,ALAM,THRSH,ERSHFT,IFLAV
C
        CLOSE(NOUNIT)
        RETURN
        END
C
        FUNCTION CEXRAT(ISP,ENRG,RLOC,SSN,DKTIME,IRDK,INRGDK,ISOLDK,
     2                  IONDK)
C
C  PROGRAMMER: R. W. SPIRO             LAST UPDATE:  05.09.90
C
C  PURPOSE:  FUNCTION SUBPROGRAM TO RETURN CHARGE EXCHANGE LOSS RATE
C         (SEC**(-1)) FOR IONS OF SPECIES ISP, ENERGY ENRG (EV) AT
C         L=RLOC (RE) FOR SUNSPOT NUMBER SSN.  THIS ROUTINE IS BASED
C         ON A TABLE GENERATED BY JAMES BISHOP OF U. OF MICHIGAN.
C
C  CALLING PARAMETERS
C        ISP       SPECIES IDENTIFIER
C                    ISP=2 FOR H+ IONS
C                    ISP=3 FOR O+ IONS
```

```
C        ENRG      ENERGY IN EV
C        RLOC      RADIAL LOCATION (RE)
C        SSN       SUNSPOT NUMBER
C        DKTIME    TABLE OF ION DECAY TIMES
C        IRDK      RADIAL DIMENSION OF DKTIME ARRAY
C        INRGDK    ENERGY DIMENSION OF DKTIME ARRAY
C        ISOLDK    SUNSPOT NUMBER DIMENSION OF DKTIME ARRAY
C        IONDK     NUMBER OF ION SPECIES IN DKTIME ARRAY
C
C
      PARAMETER (IRSIZ=18,INRGSZ=13,ISOLSZ=2,IONSIZ=2)
C
      DIMENSION DKTIME(IRDK,INRGDK,ISOLDK,IONDK)
C
      DIMENSION ELGVEC(INRGSZ), RVEC(IRSIZ),SSNVEC(2)
C
      DATA ELGVEC /2.50,2.75,3.00,3.25,3.50,3.75,4.00,
     2            4.25,4.50,4.75,5.00,5.25,5.50/
C
      DATA RVEC /1.50,2.00,2.50,3.00,
     2          3.50,4.00,4.50,5.00,
     3          5.50,6.00,6.50,7.00,
     4          7.50,8.00,8.50,9.00,
     5          9.50,10.00/
C
      DATA SSNVEC /0.0,100./
C
C
C  CHECK DIMENSIONS
C
      IF(IRSIZ.NE.IRDK.OR.INRGSZ.NE.INRGDK.OR.IONSIZ.NE.IONDK.OR.
     2   ISOLSZ.NE.ISOLDK) THEN
         WRITE(6,*) 'DIMENSION ERROR IN SUBROUTINE CEXRAT'
         WRITE(6,*) 'IRDK,INRGDK,IONDK,ISOLDK',IRDK,INRGDK,IONDK,ISOLDK
         WRITE(6,*) 'IRSIZ,INRGSZ,IONSIZ,ISOLSZ',IRSIZ,INRGSZ,IONSIZ,
     2              ISOLSZ
         WRITE(6,*) 'STOPPING PROGRAM IN CEXRAT'
         STOP
      END IF
C
C
C  WORK WITH LOG10 OF PARTICLE ENERGY
      ENRGLG=ALOG10(ENRG)
C
      ISPNDX=ISP-1
C
C  FIND BR FOR INTERPOLATION
      IF(RLOC.LE.RVEC(1)) THEN
         BR=1.
      ELSE IF(RLOC.GT.RVEC(IRDK)) THEN
         BR=IRDK
      ELSE
         DO 10 IR=1,IRDK-1
            IF(RLOC.LE.RVEC(IR+1)) THEN
               BR=IR+(RLOC-RVEC(IR))/(RVEC(IR+1)-RVEC(IR))
               GO TO 20
            END IF
   10    CONTINUE
C
      END IF
C
C
   20 CONTINUE
C
```

```
C
C  FIND BNRG FOR INTERPOLATION
      IF(ENRGLG.LE.ELGVEC(1)) THEN
         BNRG=1.
      ELSE IF(ENRGLG.GT.ELGVEC(INRGDK)) THEN
         BNRG=INRGDK
      ELSE
         DO 30 INRG=1,INRGDK-1
            IF(ENRGLG.LE.ELGVEC(INRG+1)) THEN
               BNRG=INRG+(ENRGLG-ELGVEC(INRG))/(ELGVEC(INRG+1)-
     2                                          ELGVEC(INRG))
               GO TO 40
            END IF
   30    CONTINUE
C
      END IF
C
C
   40 CONTINUE
C
C
C  FIND BSSN FOR INTERPOLATION
      BSSN=1.+(SSN-SSNVEC(1))/(SSNVEC(2)-SSNVEC(1))
C
C  DECAYT IS DECAY TIME IN SECONDS
      DECAYT=G3NTRP(DKTIME(1,1,1,ISPNDX),IRDK,INRGDK,ISOLDK,BR,BNRG,
     2             BSSN)
C
      IF(ABS(DECAYT).LT.1.E-20) THEN
         WRITE(6,*) 'DECAYT IS LESS THAN 1.E-20 SEC IN CEXRAT'
         WRITE(6,*) 'DECAYT=',DECAYT,'  BR=',BR,'  BNRG=',BNRG,'BSSN=',
     2              BSSN
         WRITE(6,*) 'ISP=',ISP,'  ENRG=',ENRG,' RLOC=',RLOC,' SSN=',SSN
      END IF
C
C  TO GET CHARGE EXCHANGE RATE (SEC**9-1)) CEXRAT, INVERT DECAYT
C
      CEXRAT=1./DECAYT
C
C  DEBUG 5.27.90
C     WRITE(6,800) ISP,ENRG,ENRGLG,RLOC,SSN
  800 FORMAT(1X,'ISP=',I2,'  ENRG=',E12.4,'  ENRGLG=',F7.4,'  RLOC=',
     2       F7.4,'  SSN=',F7.4)
C     WRITE(6,802) BR,BNRG,BSSN,DECAYT
  802 FORMAT(1X,'BR=',F7.4,'  BNRG=',F7.4,'  BSSN=',F7.4,'  DECAYT=',
     2       1PE12.4)
C
C  END DEBUG 5.27.90
C
C
C
      RETURN
      END
      SUBROUTINE READDK(IRDK,INRGDK,ISOLDK,IONDK,DKTIME)
C
C  PURPOSE:  SUBROUTINE TO READ ION CHARGE EXCHANGE DECAY TABLE
C            FURNISHED BY JAMES BISHOP OF UNIV. OF MICHIGAN
C
C  LAST UPDATE: 05.10.90
C
C  COMMON LUNIT CONTAINS LOGICAL UNIT NUMBERS FOR INPUT AND OUTPUT
         COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
```

```
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION DKTIME(IRDK,INRGDK,ISOLDK,IONDK)
C
      READ(LUDK,800) DKTIME
  800 FORMAT(8(1X,1PE9.3))
C
      RETURN
      END
      SUBROUTINE FLX2ET(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,ITM,IFLAV,
     2                  ALAM,ALMDEL,BNDLOC,VM,FLUX,ETA)
C
C  PURPOSE:  CONVERT FROM PARTICLE FLUX (PARTICLES/CM**2/S) TO
C            INVARIANT FLUX ETA
C
C  INPUT CALLING PARAMETERS
C     LATDIM       NUMBER OF LATITUDINAL GRID POINTS
C     LTDIM        NUMBER OF LONGITUDINAL GRID POINTS
C     IEDIM        MAXIMUM NUMBER OF MODEL ENERGY CHANNELS
C     IEMAX        ACTUAL NUMBER OF MODEL ENERGY CHANNELS
C     ITMDIM       MAXIMUM TIME DEPTH
C     ITM          TIME LABEL
C     IFLAV        VECTOR GIVING CHARGE AND MASS SPECIES OF PARTICLES
C                  1=ELECTRONS, 2=H+ IONS, 3=O+ IONS
C     ALAM         ENERGY INVARIANT VECTOR
C     ALMDEL       ENERGY CHANNEL DIFFERENCE VECTOR
C     BNDLOC       BOUNDARY LOCATION ARRAY
C     VM           FLUX TUBE VOLUME**(-2/3)
C     FLUX         PARTICLE NUMBER FLUX
C
C
C  OUTPUT PARAMETERS
C     ETA          INVARIANT NUMBER FLUX (PARTICLES/UNIT MAGNETIC FLUX)
C
C
      DIMENSION IFLAV(IEDIM),ALAM(IEDIM),ALMDEL(IEDIM)
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),FLUX(LATDIM,LTDIM,IEDIM)
      DIMENSION ETA(LATDIM,LTDIM,IEDIM),BNDLOC(LTDIM,ITMDIM)
C
      DO 10 IE=1,IEMAX
         IF(IFLAV(IE).EQ.2) ATOMWT=1.
         IF(IFLAV(IE).EQ.3) ATOMWT=16.
C
         DO 20 J=1,LTDIM
            IB=IFIX(BNDLOC(J,ITM))
C
            DO 30 I=IB,LATDIM
               IF(IFLAV(IE).EQ.1) THEN
                  ETA(I,J,IE)=FLUX(I,J,IE)*ALMDEL(IE)/7.392E-16/
     2                        SQRT(ABS(ALAM(IE)))/VM(I,J,ITM)
               ELSE
                  ETA(I,J,IE)=FLUX(I,J,IE)*ALMDEL(IE)*SQRT(ATOMWT)/
     2                      1.731E-17/SQRT(ABS(ALAM(IE)))/VM(I,J,ITM)
               END IF
   30       CONTINUE
C
            DO 35 I=1,IB-1
               ETA(I,J,IE)=ETA(IB,J,IE)
   35       CONTINUE
C
   20    CONTINUE
C
   10 CONTINUE
C
```

```
      RETURN
      END
      SUBROUTINE ET2FLX(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,ITM,IFLAV,
     2                  ALAM,ALMDEL,BNDLOC,VM,FLUX,ETA)
C
C  PURPOSE:  CONVERT FROM INVARIANT FLUX ETA TO FLUX IN
C            PARTICLES/CM**2/SA
C
C  INPUT CALLING PARAMETERS
C     LATDIM        NUMBER OF LATITUDINAL GRID POINTS
C     LTDIM         NUMBER OF LONGITUDINAL GRID POINTS
C     IEDIM         MAXIMUM NUMBER OF MODEL ENERGY CHANNELS
C     IEMAX         ACTUAL NUMBER OF MODEL ENERGY CHANNELS
C     ITMDIM        MAXIMUM TIME DEPTH
C     ITM           TIME LABEL
C     IFLAV         VECTOR GIVING CHARGE AND MASS SPECIES OF PARTICLES
C                   1=ELECTRONS, 2=H+ IONS, 3=O+ IONS
C     ALAM          ENERGY INVARIANT VECTOR
C     ALMDEL        ENERGY CHANNEL DIFFERENCE VECTOR
C     BNDLOC        BOUNDARY LOCATION ARRAY
C     VM            FLUX TUBE VOLUME**(-2/3)
C     FLUX          PARTICLE NUMBER FLUX
C
C
C  OUTPUT PARAMETERS
C     ETA           INVARIANT NUMBER FLUX (PARTICLES/UNIT MAGNETIC FLUX)
C
C
      DIMENSION IFLAV(IEDIM),ALAM(IEDIM),ALMDEL(IEDIM)
      DIMENSION VM(LATDIM,LTDIM,ITMDIM),FLUX(LATDIM,LTDIM,IEDIM)
      DIMENSION ETA(LATDIM,LTDIM,IEDIM),BNDLOC(LTDIM,ITMDIM)
C
      DO 10 IE=1,IEMAX
         IF(IFLAV(IE).EQ.2) ATOMWT=1.
         IF(IFLAV(IE).EQ.3) ATOMWT=16.
C
         DO 20 J=1,LTDIM
            IB=IFIX(BNDLOC(J,ITM))
C
            DO 30 I=IB,LATDIM
               IF(IFLAV(IE).EQ.1) THEN
                  FLUX(I,J,IE)=ETA(I,J,IE)*7.392E-16*VM(I,J,ITM)*
     2                         SQRT(ABS(ALAM(IE)))/ALMDEL(IE)
               ELSE
                  FLUX(I,J,IE)=ETA(I,J,IE)*VM(I,J,ITM)*1.731E-17*
     2                     SQRT(ABS(ALAM(IE)))/ALMDEL(IE)/SQRT(ATOMWT)
               END IF
   30       CONTINUE
C
            DO 35 I=1,IB-1
               FLUX(I,J,IE)=FLUX(IB,J,IE)
   35       CONTINUE
C
   20    CONTINUE
C
   10 CONTINUE
C
      RETURN
      END
```

Directory DUC0:[BALES.MSM.JULY_DELIV] LIST OF B-FIELD DAT. FILES

BO311011.DAT;1	BO311021.DAT;1	BO311031.DAT;1	BO311041.DAT;1
BO311051.DAT;1	BO311061.DAT;1	BO311071.DAT;1	BO311081.DAT;1
BO311091.DAT;1	BO311101.DAT;1	BO311111.DAT;1	BO311121.DAT;1
BO311131.DAT;1	BO311141.DAT;1	BO311151.DAT;1	BO311161.DAT;1
BO312011.DAT;1	BO312021.DAT;1	BO312031.DAT;1	BO312041.DAT;1
BO312051.DAT;1	BO312061.DAT;1	BO312071.DAT;1	BO312081.DAT;1
BO312091.DAT;1	BO312101.DAT;1	BO312111.DAT;1	BO312121.DAT;1
BO312131.DAT;1	BO312141.DAT;1	BO312151.DAT;1	BO312161.DAT;1
BO313011.DAT;1	BO313021.DAT;1	BO313031.DAT;1	BO313041.DAT;1
BO313051.DAT;1	BO313061.DAT;1	BO313071.DAT;1	BO313081.DAT;1
BO313091.DAT;1	BO313101.DAT;1	BO313111.DAT;1	BO313121.DAT;1
BO313131.DAT;1	BO313141.DAT;1	BO313151.DAT;1	BO313161.DAT;1
BO314011.DAT;1	BO314021.DAT;1	BO314031.DAT;1	BO314041.DAT;1
BO314051.DAT;1	BO314061.DAT;1	BO314071.DAT;1	BO314081.DAT;1
BO314091.DAT;1	BO314101.DAT;1	BO314111.DAT;1	BO314121.DAT;1
BO314131.DAT;1	BO314141.DAT;1	BO314151.DAT;1	BO314161.DAT;1
BO315011.DAT;1	BO315021.DAT;1	BO315031.DAT;1	BO315041.DAT;1
BO315051.DAT;1	BO315061.DAT;1	BO315071.DAT;1	BO315081.DAT;1
BO315091.DAT;1	BO315101.DAT;1	BO315111.DAT;1	BO315121.DAT;1
BO315131.DAT;1	BO315141.DAT;1	BO315151.DAT;1	BO315161.DAT;1
BO316011.DAT;1	BO316021.DAT;1	BO316031.DAT;1	BO316041.DAT;1
BO316051.DAT;1	BO316061.DAT;1	BO316071.DAT;1	BO316081.DAT;1
BO316091.DAT;1	BO316101.DAT;1	BO316111.DAT;1	BO316121.DAT;1
BO316131.DAT;1	BO316141.DAT;1	BO316151.DAT;1	BO316161.DAT;1
BO317011.DAT;1	BO317021.DAT;1	BO317031.DAT;1	BO317041.DAT;1
BO317051.DAT;1	BO317061.DAT;1	BO317071.DAT;1	BO317081.DAT;1
BO317091.DAT;1	BO317101.DAT;1	BO317111.DAT;1	BO317121.DAT;1
BO317131.DAT;1	BO317141.DAT;1	BO317151.DAT;1	BO317161.DAT;1
BO318011.DAT;1	BO318021.DAT;1	BO318031.DAT;1	BO318041.DAT;1
BO318051.DAT;1	BO318061.DAT;1	BO318071.DAT;1	BO318081.DAT;1
BO318091.DAT;1	BO318101.DAT;1	BO318111.DAT;1	BO318121.DAT;1
BO318131.DAT;1	BO318141.DAT;1	BO318151.DAT;1	BO318161.DAT;1
BO321011.DAT;1	BO321021.DAT;1	BO321031.DAT;1	BO321041.DAT;1
BO321051.DAT;1	BO321061.DAT;1	BO321071.DAT;1	BO321081.DAT;1
BO321091.DAT;1	BO321101.DAT;1	BO321111.DAT;1	BO321121.DAT;1
BO321131.DAT;1	BO321141.DAT;1	BO321151.DAT;1	BO321161.DAT;1
BO322011.DAT;1	BO322021.DAT;1	BO322031.DAT;1	BO322041.DAT;1
BO322051.DAT;1	BO322061.DAT;1	BO322071.DAT;1	BO322081.DAT;1
BO322091.DAT;1	BO322101.DAT;1	BO322111.DAT;1	BO322121.DAT;1
BO322131.DAT;1	BO322141.DAT;1	BO322151.DAT;1	BO322161.DAT;1
BO323011.DAT;1	BO323021.DAT;1	BO323031.DAT;1	BO323041.DAT;1
BO323051.DAT;1	BO323061.DAT;1	BO323071.DAT;1	BO323081.DAT;1
BO323091.DAT;1	BO323101.DAT;1	BO323111.DAT;1	BO323121.DAT;1
BO323131.DAT;1	BO323141.DAT;1	BO323151.DAT;1	BO323161.DAT;1
BO324011.DAT;1	BO324021.DAT;1	BO324031.DAT;1	BO324041.DAT;1
BO324051.DAT;1	BO324061.DAT;1	BO324071.DAT;1	BO324081.DAT;1
BO324091.DAT;1	BO324101.DAT;1	BO324111.DAT;1	BO324121.DAT;1
BO324131.DAT;1	BO324141.DAT;1	BO324151.DAT;1	BO324161.DAT;1
BO325011.DAT;1	BO325021.DAT;1	BO325031.DAT;1	BO325041.DAT;1
BO325051.DAT;1	BO325061.DAT;1	BO325071.DAT;1	BO325081.DAT;1
BO325091.DAT;1	BO325101.DAT;1	BO325111.DAT;1	BO325121.DAT;1
BO325131.DAT;1	BO325141.DAT;1	BO325151.DAT;1	BO325161.DAT;1
BO326011.DAT;1	BO326021.DAT;1	BO326031.DAT;1	BO326041.DAT;1
BO326051.DAT;1	BO326061.DAT;1	BO326071.DAT;1	BO326081.DAT;1
BO326091.DAT;1	BO326101.DAT;1	BO326111.DAT;1	BO326121.DAT;1
BO326131.DAT;1	BO326141.DAT;1	BO326151.DAT;1	BO326161.DAT;1
BO327011.DAT;1	BO327021.DAT;1	BO327031.DAT;1	BO327041.DAT;1

BO327051.DAT;1	BO327061.DAT;1	BO327071.DAT;1	BO327081.DAT;1
BO327091.DAT;1	BO327101.DAT;1	BO327111.DAT;1	BO327121.DAT;1
BO327131.DAT;1	BO327141.DAT;1	BO327151.DAT;1	BO327161.DAT;1
BO328011.DAT;1	BO328021.DAT;1	BO328031.DAT;1	BO328041.DAT;1
BO328051.DAT;1	BO328061.DAT;1	BO328071.DAT;1	BO328081.DAT;1
BO328091.DAT;1	BO328101.DAT;1	BO328111.DAT;1	BO328121.DAT;1
BO328131.DAT;1	BO328141.DAT;1	BO328151.DAT;1	BO328161.DAT;1
BO331011.DAT;1	BO331021.DAT;1	BO331031.DAT;1	BO331041.DAT;1
BO331051.DAT;1	BO331061.DAT;1	BO331071.DAT;1	BO331081.DAT;1
BO331091.DAT;1	BO331101.DAT;1	BO331111.DAT;1	BO331121.DAT;1
BO331131.DAT;1	BO331141.DAT;1	BO331151.DAT;1	BO331161.DAT;1
BO332011.DAT;1	BO332021.DAT;1	BO332031.DAT;1	BO332041.DAT;1
BO332051.DAT;1	BO332061.DAT;1	BO332071.DAT;1	BO332081.DAT;1
BO332091.DAT;1	BO332101.DAT;1	BO332111.DAT;1	BO332121.DAT;1
BO332131.DAT;1	BO332141.DAT;1	BO332151.DAT;1	BO332161.DAT;1
BO333011.DAT;1	BO333021.DAT;1	BO333031.DAT;1	BO333041.DAT;1
BO333051.DAT;1	BO333061.DAT;1	BO333071.DAT;1	BO333081.DAT;1
BO333091.DAT;1	BO333101.DAT;1	BO333111.DAT;1	BO333121.DAT;1
BO333131.DAT;1	BO333141.DAT;1	BO333151.DAT;1	BO333161.DAT;1
BO334011.DAT;1	BO334021.DAT;1	BO334031.DAT;1	BO334041.DAT;1
BO334051.DAT;1	BO334061.DAT;1	BO334071.DAT;1	BO334081.DAT;1
BO334091.DAT;1	BO334101.DAT;1	BO334111.DAT;1	BO334121.DAT;1
BO334131.DAT;1	BO334141.DAT;1	BO334151.DAT;1	BO334161.DAT;1
BO335011.DAT;1	BO335021.DAT;1	BO335031.DAT;1	BO335041.DAT;1
BO335051.DAT;1	BO335061.DAT;1	BO335071.DAT;1	BO335081.DAT;1
BO335091.DAT;1	BO335101.DAT;1	BO335111.DAT;1	BO335121.DAT;1
BO335131.DAT;1	BO335141.DAT;1	BO335151.DAT;1	BO335161.DAT;1
BO336011.DAT;1	BO336021.DAT;1	BO336031.DAT;1	BO336041.DAT;1
BO336051.DAT;1	BO336061.DAT;1	BO336071.DAT;1	BO336081.DAT;1
BO336091.DAT;1	BO336101.DAT;1	BO336111.DAT;1	BO336121.DAT;1
BO336131.DAT;1	BO336141.DAT;1	BO336151.DAT;1	BO336161.DAT;1
BO337011.DAT;1	BO337021.DAT;1	BO337031.DAT;1	BO337041.DAT;1
BO337051.DAT;1	BO337061.DAT;1	BO337071.DAT;1	BO337081.DAT;1
BO337091.DAT;1	BO337101.DAT;1	BO337111.DAT;1	BO337121.DAT;1
BO337131.DAT;1	BO337141.DAT;1	BO337151.DAT;1	BO337161.DAT;1
BO338011.DAT;1	BO338021.DAT;1	BO338031.DAT;1	BO338041.DAT;1
BO338051.DAT;1	BO338061.DAT;1	BO338071.DAT;1	BO338081.DAT;1
BO338091.DAT;1	BO338101.DAT;1	BO338111.DAT;1	BO338121.DAT;1
BO338131.DAT;1	BO338141.DAT;1	BO338151.DAT;1	BO338161.DAT;1
BO341011.DAT;1	BO341021.DAT;1	BO341031.DAT;1	BO341041.DAT;1
BO341051.DAT;1	BO341061.DAT;1	BO341071.DAT;1	BO341081.DAT;1
BO341091.DAT;1	BO341101.DAT;1	BO341111.DAT;1	BO341121.DAT;1
BO341131.DAT;1	BO341141.DAT;1	BO341151.DAT;1	BO341161.DAT;1
BO342011.DAT;1	BO342021.DAT;1	BO342031.DAT;1	BO342041.DAT;1
BO342051.DAT;1	BO342061.DAT;1	BO342071.DAT;1	BO342081.DAT;1
BO342091.DAT;1	BO342101.DAT;1	BO342111.DAT;1	BO342121.DAT;1
BO342131.DAT;1	BO342141.DAT;1	BO342151.DAT;1	BO342161.DAT;1
BO343011.DAT;1	BO343021.DAT;1	BO343031.DAT;1	BO343041.DAT;1
BO343051.DAT;1	BO343061.DAT;1	BO343071.DAT;1	BO343081.DAT;1
BO343091.DAT;1	BO343101.DAT;1	BO343111.DAT;1	BO343121.DAT;1
BO343131.DAT;1	BO343141.DAT;1	BO343151.DAT;1	BO343161.DAT;1
BO344011.DAT;1	BO344021.DAT;1	BO344031.DAT;1	BO344041.DAT;1
BO344051.DAT;1	BO344061.DAT;1	BO344071.DAT;1	BO344081.DAT;1
BO344091.DAT;1	BO344101.DAT;1	BO344111.DAT;1	BO344121.DAT;1
BO344131.DAT;1	BO344141.DAT;1	BO344151.DAT;1	BO344161.DAT;1
BO345011.DAT;1	BO345021.DAT;1	BO345031.DAT;1	BO345041.DAT;1
BO345051.DAT;1	BO345061.DAT;1	BO345071.DAT;1	BO345081.DAT;1
BO345091.DAT;1	BO345101.DAT;1	BO345111.DAT;1	BO345121.DAT;1
BO345131.DAT;1	BO345141.DAT;1	BO345151.DAT;1	BO345161.DAT;1
BO346011.DAT;1	BO346021.DAT;1	BO346031.DAT;1	BO346041.DAT;1

```
BO346051.DAT;1    BO346061.DAT;1    BO346071.DAT;1    BO346081.DAT;1
BO346091.DAT;1    BO346101.DAT;1    BO346111.DAT;1    BO346121.DAT;1
BO346131.DAT;1    BO346141.DAT;1    BO346151.DAT;1    BO346161.DAT;1
BO347011.DAT;1    BO347021.DAT;1    BO347031.DAT;1    BO347041.DAT;1
BO347051.DAT;1    BO347061.DAT;1    BO347071.DAT;1    BO347081.DAT;1
BO347091.DAT;1    BO347101.DAT;1    BO347111.DAT;1    BO347121.DAT;1
BO347131.DAT;1    BO347141.DAT;1    BO347151.DAT;1    BO347161.DAT;1
BO351011.DAT;1    BO351021.DAT;1    BO351031.DAT;1    BO351041.DAT;1
BO351051.DAT;1    BO351061.DAT;1    BO351071.DAT;1    BO351081.DAT;1
BO351091.DAT;1    BO351101.DAT;1    BO351111.DAT;1    BO351121.DAT;1
BO351131.DAT;1    BO351141.DAT;1    BO351151.DAT;1    BO351161.DAT;1
BO352011.DAT;1    BO352021.DAT;1    BO352031.DAT;1    BO352041.DAT;1
BO352051.DAT;1    BO352061.DAT;1    BO352071.DAT;1    BO352081.DAT;1
BO352091.DAT;1    BO352101.DAT;1    BO352111.DAT;1    BO352121.DAT;1
BO352131.DAT;1    BO352141.DAT;1    BO352151.DAT;1    BO352161.DAT;1
BO353011.DAT;1    BO353021.DAT;1    BO353031.DAT;1    BO353041.DAT;1
BO353051.DAT;1    BO353061.DAT;1    BO353071.DAT;1    BO353081.DAT;1
BO353091.DAT;1    BO353101.DAT;1    BO353111.DAT;1    BO353121.DAT;1
BO353131.DAT;1    BO353141.DAT;1    BO353151.DAT;1    BO353161.DAT;1
BO354011.DAT;1    BO354021.DAT;1    BO354031.DAT;1    BO354041.DAT;1
BO354051.DAT;1    BO354061.DAT;1    BO354071.DAT;1    BO354081.DAT;1
BO354091.DAT;1    BO354101.DAT;1    BO354111.DAT;1    BO354121.DAT;1
BO354131.DAT;1    BO354141.DAT;1    BO354151.DAT;1    BO354161.DAT;1
BO355011.DAT;1    BO355021.DAT;1    BO355031.DAT;1    BO355041.DAT;1
BO355051.DAT;1    BO355061.DAT;1    BO355071.DAT;1    BO355081.DAT;1
BO355091.DAT;1    BO355101.DAT;1    BO355111.DAT;1    BO355121.DAT;1
BO355131.DAT;1    BO355141.DAT;1    BO355151.DAT;1    BO355161.DAT;1
BO356011.DAT;1    BO356021.DAT;1    BO356031.DAT;1    BO356041.DAT;1
BO356051.DAT;1    BO356061.DAT;1    BO356071.DAT;1    BO356081.DAT;1
BO356091.DAT;1    BO356101.DAT;1    BO356111.DAT;1    BO356121.DAT;1
BO356131.DAT;1    BO356141.DAT;1    BO356151.DAT;1    BO356161.DAT;1
BO357011.DAT;1    BO357021.DAT;1    BO357031.DAT;1    BO357041.DAT;1
BO357051.DAT;1    BO357061.DAT;1    BO357071.DAT;1    BO357081.DAT;1
BO357091.DAT;1    BO357101.DAT;1    BO357111.DAT;1    BO357121.DAT;1
BO357131.DAT;1    BO357141.DAT;1    BO357151.DAT;1    BO357161.DAT;1
DIRECTORY.LIS;1

Total of 609 files.
```

Appendix C

Appendix C is attached separately as a computer listing.

Appendix D

```
$define sys$input tt
!$define sys$output tt
$define bfield [-.bfields]
$define for008 msmout.dat
$define for014 ashift.dat
$define for015 aflux.dat
$define for016 avm.dat
$define for017 abndloc.dat
$define for018 axmin.dat
$define for019 aymin.dat
$define for020 azmin.dat
$define for044 aupdat.dat
$run mapint
```

```
      PROGRAM MSMMAP
C                 VERSION OF 4/23/90
C                 PROGRAMMED BY R. V. HILMER
C                 REQUIRES LINKAGE WITH BFIELD MODEL SUBROUTINES,
C                 SUBROUTINES FNDBRK, AND B-MATRIX DATA FILES
C
C     PURPOSE-    THIS PROGRAM COLLECTS MAGNETOSPHERIC SPECIFICATION
C                 MODEL INPUTS AND AN ARBITRARY GSM STARTING POINT,
C                 VALIDATES THE INPUTS AND FINDS THE POINT ON THE MSM
C                 CALCULATION SURFACE WHICH IS MAGNETICALLY CONNECTED
C                 TO THE GSM STARTING POINT.
C                 IT THEN INTERPOLATES IN SPACE, TIME AND ENERGY TO
C                 GET THE FLUXES AT THE MAPPED POINT.
C     VARIABLES-  FTILT: DIPOLE TILT ANGLE (DEGREES, POSITIVE FOR
C                        NORTHERN HEMISPHERE SUMMER)
C                 FSTDFF: MAGNETOPAUSE STAND-OFF DISTANCE (RE)
C                 FDST:  MAGNETIC ACTIVITY INDEX DST (NT)
C                 FEQEDG: DIPOLE LAT. OF MIDNIGHT EQUATORWARD DIFFUSE
C                        AURORA BOUNDARY (DEG).
C                 FCLPSE: MAGNETOTAIL FIELD COLLAPSE INDICATOR
C                 XSTRT,YSTRT,ZSTRT: MAGNETIC TRACING START POINT (GSM)
C                 IVALID: INDICATES IF INSERTED INPUTS ARE WITHIN VALID
C                         RANGES. (1 = YES, 0 = NO)
C                 XMAP,YMAP,ZMAP: MAGNETIC MAPPING POINT ON THE MSM
C                         COMPUTATIONAL SURFACE (GSM).
C     SUBROUTINES-MAPTRK: DETERMINES THE MAGNETIC FIELD MAPPING
C                    CONNECTING AN ARBITRARY POINT IN SPACE TO THE MSM
C                    COMPUTIONAL SURFACE. USES INTERPOLATION SCHEME
C                    INVOLVING PRECALCULATED INTERNAL MODEL PARAMETER
C                    SETS TO FIND THE MAPPING FOR ARBITRARY MSM INPUT.
C                 VALID: CHECKS TO SEE IF EXTERNAL INPUT PARAMETERS
C                        AND TRACING START POINT ARE VALID.
C                 READ3D:  READS DATA FILES FOR FLUX CALCULATION.
C                 LOCATE:  FINDS THE INTERPOLATED POINT ON A
C                          2-DIMENSIONAL GRID.
C                 CHECK:  CHECKS THAT THE INPUT TIMES ARE REASONABLE.
C                 TCHECK:  LOCATES THE 2 RECORDS WHICH SPAN THE TIME
C                          REQUESTED.
C                 RDVEC:  READS THE HEADER VECTOR OF THE FLUX FILES
C                 SETALM:  SETS UP THE INVARIANT ENERGY ARRAY
C      FUNCTIONS:  G3NTRP:  3-DIMENSIONAL INTERPOLATION ROUTINE

       PARAMETER (LATDIM=62,LTDIM=51,IEDIM=30,ITMDIM=50, NAUGEL=28)
       DIMENSION R(LATDIM,LTDIM),BNDLOC(LTDIM,2),P(LATDIM,LTDIM),
     1  VM(LATDIM,LTDIM,2),FLUX(LATDIM,LTDIM,IEDIM),ALAM(IEDIM),
     2  ALMDEL(IEDIM),ID(20),RID(20),THRSH(IEDIM),ERSHFT(IEDIM),
     3  XMIN(LATDIM,LTDIM,2),YMIN(LATDIM,LTDIM,2),ZMIN(LATDIM,LTDIM,2),
     4  AUGPAR(NAUGEL),AGPAR1(NAUGEL),FLUX1(LATDIM,LTDIM,IEDIM),
     5  ISTART(3),IFLAV(IEDIM),FLXOUT(2)
       CHARACTER*80 CHID

      PI=ATAN2(0.,-1.)
C
C  THE INTERACTIVE PORTION OF THE PROGRAM CHECKS FOR CORRECT TIME
C   ENTRIES THROUGH SUBROUTINE CHECK.
C
100   WRITE(6,*)' ENTER LAST 2 DIGITS OF YEAR AND JULIAN DAY '
      READ(5,*)IYEAR,IDAY
      WRITE(6,*)' ENTER HOUR, MINUTE, AND SECONDS '
      READ(5,*)IHOUR,MINUTE,ISCNDS
      CALL CHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,IFLAG)
      IF(IFLAG.EQ. 1)THEN
       WRITE(6,*)
       WRITE(6,*)' INCORRECT INPUT.  DO YOU WANT TO TRY AGAIN?'
```

```
C     WRITE(6,*)'ALAM',ALAM,'THRSH',THRSH,'ERSHFT',ERSHFT,'IFLAV',IFLAV

      CALL READ3D(15,'FLUX   ',IREC,LATDIM,LTDIM,IEMAX,ITMDIM,ID,RID,
     1   CHID,FLUX)
      CALL READ3D(15,'FLUX ',IREC+1,LATDIM,LTDIM,IEMAX,ITMDIM,ID,
     1   RID,CHID,FLUX1)

      IF((ID(8).NE.LATDIM).OR.(ID(9).NE.LTDIM))THEN
       WRITE(6,*)
       WRITE(6,*)' INCORRECT PARAMETERS FOR GRID-STOPPING PROGRAM'
       WRITE(6,*)' ID(8) = ',ID(8),' LATDIM = ',LATDIM
       WRITE(6,*)' ID(9) = ',ID(9),' LTDIM  = ',LTDIM
       WRITE(6,*)
       STOP
      ENDIF
      CALL READ3D(16,'VM     ',IREC,LATDIM,LTDIM,1,ITMDIM,ID,RID,CHID,
     1  VM(1,1,1))
      CALL READ3D(16,'VM     ',IREC+1,LATDIM,LTDIM,1,ITMDIM,ID,RID,CHID,
     1  VM(1,1,2))
      CALL READ3D(17,'BNDLOC',IREC,LTDIM,1,1,ITMDIM,ID,RID,CHID,
     1  BNDLOC(1,1))
      CALL READ3D(17,'BNDLOC',IREC+1,LTDIM,1,1,ITMDIM,ID,RID,CHID,
     1  BNDLOC(1,2))
      CALL READ3D(18,'XMIN  ',IREC,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,XMIN(1,1,1))
      CALL READ3D(18,'XMIN  ',IREC+1,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,XMIN(1,1,2))
      CALL READ3D(19,'YMIN  ',IREC,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,YMIN(1,1,1))
      CALL READ3D(19,'YMIN  ',IREC+1,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,YMIN(1,1,2))
      CALL READ3D(20,'ZMIN  ',IREC,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,ZMIN(1,1,1))
      CALL READ3D(20,'ZMIN  ',IREC+1,LATDIM,LTDIM,1,ITMDIM,ID,RID,
     1  CHID,ZMIN(1,1,2))
C
C  SET UP THE TIME INTERPOLATION FOR THE B-FIELD COMPONENTS AND
C   CALCULATE RADIUS AND LOCAL TIME ARRAYS
C
      TIME=(FLOAT(IHOUR)*3600.0) + (FLOAT(MINUTE)*60.0) + FLOAT(ISCNDS)
      TIME1=(AUGPAR(3)*3600.0) + (AUGPAR(4)*60.0) + AUGPAR(5)
      TIME2=(AGPAR1(3)*3600.0) + (AGPAR1(4)*60.0)  + AGPAR1(5)
      TIMTRP =((TIME-TIME1)/(TIME2-TIME1)) + 1.0
C     WRITE(6,*)'TIME,TIME1,TIME2,TIMTRP',TIME,TIME1,TIME2,TIMTRP
      DO 400 I=1,LATDIM
       DO 450 J=1,LTDIM
        XNTRP=G3NTRP(XMIN,LATDIM,LTDIM,2,FLOAT(I),FLOAT(J),TIMTRP)
        YNTRP=G3NTRP(YMIN,LATDIM,LTDIM,2,FLOAT(I),FLOAT(J),TIMTRP)
        ZNTRP=G3NTRP(ZMIN,LATDIM,LTDIM,2,FLOAT(I),FLOAT(J),TIMTRP)
        R(I,J)=SQRT(XNTRP**2 + YNTRP**2 + ZNTRP**2)
        IF((XNTRP.EQ.0.0).AND.(YNTRP.EQ.0.0))THEN
         P(I,J)=0.0
        ELSE
         P(I,J)=ATAN2(YNTRP,XNTRP)
         IF(P(I,J).LT.0.0)P(I,J)=P(I,J)+ 2.0*PI
        ENDIF
450    CONTINUE
400   CONTINUE
C     WRITE(6,*)' IREC ',IREC
C
C  SET UP INVARIANT ARRAYS FOR THE FLUX CALCULATIONS
C
      CALL SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)
 10   CONTINUE
```

```
        WRITE(6,*)' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN.'
        READ(5,*)IANSW
        WRITE(6,*)
        IF(IANSW.EQ.0)STOP
        GO TO 100
       ENDIF
       IREC=1
C
C  THIS PORTION OF THE CODE SEARCHES THE OUTPUT DATA BASE TO FIND THE
C   2 TIMES CLOSEST TO THE INPUT TIME.
C
200    CALL READ3D(44,'UPDATE',IREC,NAUGEL,1,1,ITMDIM,ID,RID,CHID,
     1  AUGPAR)
C      WRITE(6,*)' IREC, ID ',IREC,ID
       IF(IREC.EQ.1)THEN
        ISTART(1)=AUGPAR(1)
        ISTART(2)=AUGPAR(2)
        ISTART(3)=AUGPAR(3)*3600 + AUGPAR(4)*60 + AUGPAR(5)
        ITMMAX=ID(10)
        IEMAX=ID(9)
       ENDIF
       IF(IREC+1.GT.ITMMAX)THEN
        WRITE(6,*)
        WRITE(6,*)' END OF FILE ENCOUNTERED WITHOUT FINDING THE TIME',
     1 ' YOU REQUESTED.'
        WRITE(6,*)' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN.'
        READ(5,*)IANSW
        IF(IANSW.EQ.0)STOP
        GO TO 100
       ENDIF
       CALL READ3D(44,'UPDATE',IREC+1,NAUGEL,1,1,ITMDIM,ID,RID,CHID,
     1  AGPAR1)
       CALL TCHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,AUGPAR,AGPAR1,
     1  NAUGEL,IFLAG,ISTART)
       IF(IFLAG.EQ.1)THEN
         IREC=IREC+1
         GO TO 200
       ENDIF
C
C  INPUT SATELLITE ENERGY IN KEV.  THIS IS CHECK TO BE GREATER THAN 0
C   AND CONVERTED TO INVARIANT ENERGY, ALAM, THE PROGRAMS BASIC
C   ENERGY UNIT.  ALSO PARTICLE TYPE IS REQUESTED AND CHECKED TO BE
C   BETWEEN 1 AND 3 (1=ELECTRONS,2=H+,3=O+)
C
300    WRITE(6,*)' INPUT ENERGY (KEV) AND 1 FOR ELECTRONS, 2 FOR H+,'
       WRITE(6,*)' OR 3 FOR O+'
       READ(5,*)ENERGY,JFLAV
       IF((JFLAV.LT.1).OR.(JFLAV.GT.3))THEN
        WRITE(6,*)
        WRITE(6,*)' INCORRECT INPUT FOR PARTICLE TYPE. PLEASE RE-ENTER.'
        WRITE(6,*)
        GO TO 300
       ENDIF
       IF(ENERGY.LE.0)THEN
        WRITE(6,*)
        WRITE(6,*)' ENERGY LESS THAN 0.  PLEASE RE-ENTER.'
        WRITE(6,*)
        GO TO 300
       ENDIF
C
C  READ INPUT FILES NEEDED FOR INTERPOLATING
C
       CALL RDVEC (14,IREC,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
```

```
       ENDIF
 550  CONTINUE
      IF(XLO.EQ.-1.0)THEN
       WRITE(6,*)
       WRITE(6,*)' ENERGY NOT CALCULATED FOR THIS POINT IN THIS RUN.'
       WRITE(6,*)
       WRITE(6,*)' FOLLOWING ARE THE ENERGIES AND SPECIES FOR THIS RUN'
       WRITE(6,*)
       DO 475 IE=1,IEMAX
        WRITE(6,*)'  ENERGY (KEV) = ',ABS(ALAM(IE)*CALVM)/1000.0,
     1    ' SPECIES NUMBER  = ',IFLAV(IE)
 475   CONTINUE
       WRITE(6,*)' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN'
       READ(5,*)IANSW
       IF(IANSW.EQ.0)STOP
       WRITE(6,*)
       GO TO 300
      ENDIF
      IF((ALAM(IFIX(BK)).LT.0.0).AND.(JFLAV.NE.1))THEN
       WRITE(6,*)
       WRITE(6,*)' PARTICLE TYPE AND INVARIANT ENERGY ARE INCORRECT.'
       WRITE(6,*)' ENTER 0 STOP, AND OTHER NUMBER TO CONTINUE.'
       READ(5,*) IANSW
       WRITE(6,*)
       IF(IANSW.EQ.0)STOP
       GO TO 300
      ENDIF
C
C  INTERPOLATE FLUX ARRAYS IN SPACE, TIME AND ENERGY
C
C
      FLXOUT(1)=G3NTRP(FLUX,LATDIM,LTDIM,IEMAX,BI,BJ,BK)
      FLXOUT(2)=G3NTRP(FLUX1,LATDIM,LTDIM,IEMAX,BI,BJ,BK)
      SATFLX=G3NTRP(FLXOUT,2,1,1,TIMTRP,1.,1.)
C     WRITE(6,*)' BI,BJ,BK,TIMTRP,FLXOUT ',BI,BJ,BK,TIMTRP,FLXOUT
C
C  WRITE OUT FLUX, INCLUDING ERROR AND THRESHOLD IF AVAILABLE
C
      ERROR=G3NTRP(ERSHFT,IEMAX,1,1,BK,1.,1.)
      WRITE(6,*)
      WRITE(6,*)' FLUX AT THE SATELLITE (LOG(#/CM2-S-KEV-STER)) = ',
     1   SATFLX
      WRITE(6,*)
      IF(ERROR.NE.0.0)THEN
       WRITE(6,*)' THE ERROR BETWEEN THE MODEL OUTPUT AND SATELLITE '
       WRITE(6,*)' DATA IS ',ERROR
       WRITE(6,*)' PLEASE NOTE THAT THE ERROR WAS CALCULATED USING '
       WRITE(6,*)' DATA AT GEOSYNCHRONOUS ORBIT AND IS ONLY VALID FOR'
       WRITE(6,*)' POINTS AT 6.6 RE.'
       WRITE(6,*)
      ENDIF
      THRSHK=G3NTRP(THRSH,IEMAX,1,1,BK,1.,1.)
      WRITE(6,*)
      IF(THRSHK.NE.0.0)THEN
       WRITE(6,*)' THE PARTICLES USED IN THE FLUX CALCULATION WERE '
       WRITE(6,*)' ABOVE THE THRESHOLD LEVEL AND THE FULL-TRACEBACK'
       WRITE(6,*)' METHOD WAS NOT USED.'
      ENDIF
C     WRITE(6,*)' ERROR, THRSHK ',ERROR,THRSHK
      WRITE(6,*)
 500  WRITE(6,*) 'TRACE FROM ANOTHER STARTING POINT'
      WRITE(6,*) 'WITH SAME TIME  PARAMETERS?       (1 = YES)'
      READ(5,*) KK
      IF (KK.EQ.1) GOTO 300
```

```
C
C  INPUT FOR SPATIAL COORDINATE OF SATELLITE
C
      FTILT= AUGPAR(23)
      FSTDFF=AUGPAR(24)
      FDST=AUGPAR(7)
      FEQEDG=AUGPAR(8)
      FCLPSE=AUGPAR(15)
 20   CONTINUE
      WRITE(6,*) 'ENTER SPATIAL POINT OF INTEREST IN GSM (RE): X,Y,Z'
      READ(5,*) XSTRT,YSTRT,ZSTRT
C       CHECK VALIDITY OF INPUT PARAMETERS AND TRACER START POINT
      CALL VALID(FTILT,FSTDFF,FDST,FEQEDG,FCLPSE,XSTRT,YSTRT,ZSTRT,
     1            IVALID)
      IF (IVALID.EQ.0) THEN
        WRITE(6,*) 'INVALID INPUT.  TRY AGAIN? (0 = YES)'
        READ(5,*) KK
        IF (KK.EQ.0) GOTO 20
        STOP
      END IF
      CALL MAPTRK(FSTDFF,FTILT,FEQEDG,FDST,FCLPSE,XSTRT,YSTRT,ZSTRT,
     1               XMAP,YMAP,ZMAP)
      IF(ABS(XMAP).LT.1.0.AND.ABS(YMAP).LT.1.0.AND.ABS(ZMAP).LT.1.0)
     1  GO TO 500
      WRITE(6,*)
      WRITE(6,*) 'INTERPOLATED MAPPING POINT: XGSM   YGSM   ZGSM'
      WRITE(6,*) XMAP,YMAP,ZMAP
      WRITE(6,*)
C
C  CHECK THAT THE MAPPED POINT IS WITHIN THE BOUNDARY OF THE MODEL
C   CALCULATION
C
      RTEST=SQRT(XMAP**2 + YMAP**2 + ZMAP**2)
      PTEST=ATAN2(YMAP,XMAP)
      IF(PTEST.GT.2.0*PI)PTEST=PTEST-(2.0*PI)
      IF(PTEST.LT.0.0)PTEST=PTEST + 2.0*PI
      CALL LOCATE(RTEST,PTEST,LATDIM,LTDIM,R,P,BI,BJ)
      JPLACE=IFIX(BJ)
      BNDY=G3NTRP(BNDLOC,LTDIM,2,1,BJ,TIMTRP,1.0)
      IPLACE=IFIX(BNDY)
      IF((R(IPLACE,JPLACE).LT.RTEST))THEN
       WRITE(6,*)
       WRITE(6,*)' POINT MAPPED TO IS OUT OF THE BOUNDARY OF THE '
       WRITE(6,*)' MSM CALCULATION REGION. '
       WRITE(6,*)' BOUNDARY LOCATION ',R(IPLACE,JPLACE),'RE'
       WRITE(6,*)' SATELLITE MAPS TO ',RTEST,'RE'
       WRITE(6,*)
       GO TO 500
      ENDIF
      CALVM=G3NTRP(VM,LATDIM,LTDIM,2,BI,BJ,TIMTRP)
      ALMCHK=(ENERGY/CALVM)*1000.0
C     WRITE(6,*)'ENERGY,CALVM,ALMCHK',ENERGY,CALVM,ALMCHK
C     WRITE(6,*)' VM(IFIX(BI,BJ)) ',VM(IFIX(BI),IFIX(BJ),1)
C
C  CHECK TO SEE IF THE ENERGY AND SPECIES INPUT IS WITHIN OUR
C    CALCULATED RANGE
C
      XLO=-1.0
      DO 550 IE=1,IEMAX-1
       IF((IFLAV(IE).NE.JFLAV).AND.(IFLAV(IE+1).NE.JFLAV))GO TO 550
       IF((ALMCHK.GE.ABS(ALAM(IE))).AND.(ALMCHK.LE.ABS(ALAM(IE+1)))
     1  .AND.(IFLAV(IE).EQ.IFLAV(IE+1)))THEN
        XLO=(ALMCHK-ABS(ALAM(IE)))/(ABS(ALAM(IE+1))-ABS(ALAM(IE)))
        BK=FLOAT(IE)+XLO
```

```
C
C     PURPOSE    -DETERMINE THE MAGNETIC FIELD MAPPING CONNECTING AN
C                 ARBITRARY POINT IN SPACE TO THE MSM COMPUTATIONAL
C                 SURFACE (THE GSM EQUATORIAL PLANE FOR DIPOLE TILT ANGLE
C                 SET EQUAL TO ZERO) MAPPINGS ARE DONE FROM A GIVEN POINT
C                 TO A SURFACE FOR ALL MODELS IMMEDIATELY ADJACENT IN
C                 THE MODELS EXTERNAL PARAMETER SPACE.  (SEE OUTLINE OF
C                 METHOD GIVEN IN MSM QUARTERLY REPORT NO. 11)
C        ***  NOTE: PROCEDURE IS STOPPED IF AN OPEN FIELD-LINE IS FOUND
C                 DURING ONE OF THE TRACES OR IF THERE ARE LESS THAN THE
C                 MINIMUM NUMBER (8) OF ESTABLISHED MODELS REQUIRED FOR
C                 A PROPER INTERPOLATION.
C     INPUT      -FTILT: DIPOLE TILT ANGLE (DEGREES, POSITIVE FOR
C                        NORTHERN HEMISPHERE SUMMER.
C                 FSTDFF: MAGNETOPAUSE STAND-OFF DISTANCE (RE)
C                 FDST: MAGNETIC ACTIVITY DST (NT)
C                 FEQEDG: DIPOLE LAT. OF MIDNIGHT EQUATORWARD DIFFUSE
C                         AURORA BOUNDARY (DEG)
C                 FCLPSE: MAGNETOTAIL FIELD COLLAPSE INDICATOR
C                 XSTRT,YSTRT,ZSTRT: STARTING POINT FOR MAGNETIC FIELD
C                                    TRACES (GSM COORDINATES)
C     OUTPUT     -XMAP,YMAP,ZMAP:  INTERPOLATED MAPPING POSITION (GSM)
C     VARIABLES  -XHIT,YHIT,ZHIT:  MAPPING POINT OF INDIVIVUAL TRACES
C                 IOPEN:  OPEN/CLOSED FIELD-LINE INDICATOR 1 = OPEN
C                                                          0 = CLOSED
C                 BFEXST: INDICATES IF B-FIELD MATRIX EXISTS
C                 TPRM: MAGNETIC FIELD MODEL INTERNAL INPUT PARAMETERS
C                 BFLIM: ESTABLISHED PARAMETERS WHICH BRACKET BFPAR
C                 T: COEFFICIENTS FOR SINGLE LINEAR INTERPOLATION USED TO
C                    CONSTRUCT VALUES OF VARIABLE COEFF.
C                 BFPAR: THE INPUT PARAMETERS
C                 COEFF: INTERPOLATION COEFFICIENT
C                 ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,DELXC,
C                     DYC,BRN,BRP,RN,RP: MAGNETIC FIELD MODEL INTERNAL
C                     INPUTS (SEE SUBROUTINE BFIELD FOR DETAILS)
C                 XMAX,YMAX,ZMAX: MAXIMUM VALUES OF XHIT,YHIT,ZHIT
C                 XMIN,YMIN,ZMIN: MINIMUM VALUES OF XHIT,YHIT,ZHIT
C                 XRANGE,YRANGE,ZRANGE: = MAXIMUM - MINIMUM OF ABOVE
C     SUBROUTINE-BFIELD: MAGNETIC FIELD MODEL POINT CALCULATIONS
C                 BMNMAP: MAPS SINGLE FIELD LINE TO MINIMUM |B| SURFACE
C                 GETIP:  COLLECTS INTERNAL BFIELD MODEL INPUTS
C                         FROM THE RESPECTIVE MATRICES FOR THE GROUP
C                         OF MODELS IN CLOSEST PROXIMITY OF PARAMETER
C                         SPACE TO THE MSM INPUTS.
C
C  NOTE: THIS SUBROUTINE CONTAINS PARTS WHICH WERE ADAPTED DIRECTLY
C        FROM SUBROUTINE BTRACE AS PROGRAMMED BY R.W. SPIRO. THESE
C        INCLUDE THE MATRIX READING PROCEDURE, DEVELOPING THE PARAMETER
C        WEIGHTING COEFFICIENTS, AND THE FIVE DIMENSIONAL INTERPOLATION
C        SCHEME.
C  REFERENCE:
C        FIVE DIMENSIONAL INTERPOLATION IS BASED ON MULTILINEAR
C          INTERPOLATION SCHEME GIVEN BY PRESS ET AL. IN
C          NUMERICAL RECIPES, CAMBRIDGE UNIVERSITY PRESS, 1987,PP 95-97
C
      LOGICAL*1 BFEXST(2,2,2,2,2)
C
      DIMENSION TPRM(14,2,2,2,2,2)
      DIMENSION XHIT(2,2,2,2,2),YHIT(2,2,2,2,2),ZHIT(2,2,2,2,2)
      DIMENSION BFLIM(2,5),BFPAR(5)
      DIMENSION T(2,5),N(5)
C
C  SET UP BFPAR VECTOR FOR CALLING GETMAT
      BFPAR(1)=FSTDFF
```

```
      WRITE(6,*) 'AGAIN WITH NEW TIME?                    (1 = YES)'
      READ(5,*) KK
      IF (KK.EQ.1) GOTO 100
C
      STOP
      END
C
C******************************** VALID
      SUBROUTINE VALID(TILT,STDF,DST,EQBDY,COLLAP,XSTRT,YSTRT,ZSTRT,
     1                 IVALID)
C
C     PURPOSE- DETERMINE IF EXTERIOR MODEL INPUTS ARE WITHIN VALID
C              RANGES.
C     INPUT-   TILT:   DIPOLE TILT ANGLE (DEG)
C              STDF:   MAGNETOPAUSE STAND-OFF DISTANCE (RE)
C              DST:    MAGNETIC FIELD STRENGTH INDEX (NT)
C              EQBDY:  MIDNIGHT EQUATORWARD BOUNDARY OF DIFFUSE
C                      AURORA (DEGREES DIPOLE LATITUDE)
C              COLLAP: SUBSTORM MAGNETOTAIL FIELD COLLAPSE INDICATOR
C              XSTRT,YSTRT,ZSTRT: GSM STARTING POINT FOR B-FIELD TRACE
C     OUTPUT-  IVALID: INDICATOR OF PARAMETER SET VALIDITY
C                      1 = VALID, 0 = INVALID
C     VARIABLE-DIST:   GEOCENTRIC DISTANCE TO TRACER STARTING POINT (RE)
C
C       SET MAXIMUM(MX) AND MINIMUM(MN) ACCEPTABLE PARAMETER VALUES
      DATA TILTMX,TILTMN/35.00,-35.00/
      DATA STDFMX,STDFMN/14.00,6.00/
      DATA DSTMX,DSTMN/50.00,-400.00/
      DATA EQBDYMX,EQBDYMN/69.30,49.47/
      DATA COLLAPMX,COLLAPMN/1.00,0.00/
C
      IVALID = 1
      IF ((TILT.LT.TILTMN).OR.(TILT.GT.TILTMX)) THEN
        IVALID = 0
        WRITE(6,*) 'TILT VALUE',TILT,'IS OUT OF RANGE'
      END IF
      IF ((STDF.LT.STDFMN).OR.(STDF.GT.STDFMX)) THEN
        IVALID = 0
        WRITE(6,*) 'STAND-OFF VALUE',STDF,'IS OUT OF RANGE'
      END IF
      IF ((DST.LT.DSTMN).OR.(DST.GT.DSTMX)) THEN
        IVALID = 0
        WRITE(6,*) 'DST VALUE',DST,'IS OUT OF RANGE'
      END IF
      IF ((EQBDY.LT.EQBDYMN).OR.(EQBDY.GT.EQBDYMX)) THEN
        IVALID = 0
        WRITE(6,*) 'EQEDGE VALUE',EQBDY,'IS OUT OF RANGE'
      END IF
      IF ((COLLAP.LT.COLLAPMN).OR.(COLLAP.GT.COLLAPMX)) THEN
        IVALID = 0
        WRITE(6,*) 'COLLAPSE VALUE',COLLAP,'IS OUT OF RANGE'
      END IF
C
      DIST = SQRT(XSTRT**2 + YSTRT**2 + ZSTRT**2)
      IF (DIST.LE.1.) THEN
        IVALID = 0
        WRITE(6,*) 'INVALID STARTING POINT FOR FIELD-LINE TRACE (<1 RE)'
      END IF
C
      RETURN
      END
C**************************  MAPTRK
      SUBROUTINE MAPTRK(FSTDFF,FTILT,FEQEDG,FDST,FCLPSE,
     1                  XSTRT,YSTRT,ZSTRT,XMAP,YMAP,ZMAP)
```

```
      BFPAR(2)=FTILT
      BFPAR(3)=FEQEDG
      BFPAR(4)=FDST
      BFPAR(5)=FCLPSE
C
C  INITIALIZE "HIT" ARRAYS AND MAPPING RESULT
      X = 1.
      Y = 1.
      Z = 1.
      XMAP = 0.
      YMAP = 0.
      ZMAP = 0.
      DO 15 I=1,2
        DO 16 J=1,2
          DO 17 K=1,2
            DO 18 L=1,2
              DO 19 M=1,2
                XHIT(I,J,K,L,M) = 0.
                YHIT(I,J,K,L,M) = 0.
                ZHIT(I,J,K,L,M) = 0.
   19         CONTINUE
   18       CONTINUE
   17     CONTINUE
   16   CONTINUE
   15 CONTINUE
C
C  GET INTERNAL B-FIELD PARAMETERS
      CALL GETIP(BFPAR,TPRM,BFLIM,BFEXST)
C
C  SET UP WEIGHTING PARAMETERS FOR INTERPOLATION
C
         DO 10 II=1,5
            T(2,II)=(BFPAR(II)-BFLIM(1,II))/(BFLIM(2,II)-BFLIM(1,II))
            T(1,II)=1.-T(2,II)
   10    CONTINUE
C
C       FIND MINUMUM CROSSING POINT FOR EACH OF THE "CORNERS"
      DO 31 I = 1,2
        DO 32 J = 1,2
          DO 33 K = 1,2
            DO 34 L = 1,2
              DO 35 M = 1,2
                IF (.NOT.BFEXST(I,J,K,L,M)) THEN
C                   DO NOT ATTEMPT A FIELD-LINE TRACE
                  GOTO 36
                END IF
C                 SET INTERNAL PARAMETERS
                STAND  = TPRM( 1,I,J,K,L,M)
                TILT   = TPRM( 2,I,J,K,L,M)
                HJNEAR = TPRM( 3,I,J,K,L,M)
                XNEAR  = TPRM( 4,I,J,K,L,M)
                HJFRAC = TPRM( 5,I,J,K,L,M)
                DY     = TPRM( 6,I,J,K,L,M)
                D      = TPRM( 7,I,J,K,L,M)
                COLLAP = TPRM( 8,I,J,K,L,M)
                DELXC  = TPRM( 9,I,J,K,L,M)
                DYC    = TPRM(10,I,J,K,L,M)
                BRN    = TPRM(11,I,J,K,L,M)
                BRP    = TPRM(12,I,J,K,L,M)
                RN     = TPRM(13,I,J,K,L,M)
                RP     = TPRM(14,I,J,K,L,M)
C
C         SET-UP BFIELD INTERNAL COEFFICIENTS AND TAIL CURRENTS
        ISWIT = 0
```

```
      CALL BFIELD(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP05080
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)       MAP05090
      CALL BFIELD(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP05100
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)       MAP05110
C                                                                       MAP05120
C       SINGLE MAGNETIC FIELD-LINE TRACE TO MINIMUM |B| SURFACE         MAP05130
      CALL BMNMAP(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,     MAP05140
     1            DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XSTRT,YSTRT,ZSTRT,     MAP05150
     2         IOPEN,XHIT(I,J,K,L,M),YHIT(I,J,K,L,M),ZHIT(I,J,K,L,M))   MAP05160
C                                                                       MAP05170
      IF (IOPEN.EQ.1) THEN                                              MAP05180
        WRITE(6,*) 'OPEN FIELD-LINE FOUND, INTERPOLATION INVALIDATED'   MAP05190
        WRITE(6,*) 'EXITED FROM SUBROUTINE MAPTRACE'                    MAP05200
        RETURN                                                          MAP05210
      END IF                                                            MAP05220
 36           CONTINUE                                                  MAP05230
 35         CONTINUE                                                    MAP05240
 34       CONTINUE                                                      MAP05250
 33     CONTINUE                                                        MAP05260
 32   CONTINUE                                                          MAP05270
 31 CONTINUE                                                            MAP05280
C                                                                       MAP05290
C     INITIALIZE RANGE SPREAD INDICATORS                                MAP05300
    XMAX = -999.                                                        MAP05310
    YMAX = -999.                                                        MAP05320
    ZMAX = -999.                                                        MAP05330
    XMIN =  999.                                                        MAP05340
    YMIN =  999.                                                        MAP05350
    ZMIN =  999.                                                        MAP05360
C     MAIN INTERPOLATION LOOPS                                          MAP05370
    DO 21 N1=1,2                                                        MAP05380
       N(1)=N1                                                          MAP05390
       DO 22 N2=1,2                                                     MAP05400
          N(2)=N2                                                       MAP05410
          DO 23 N3=1,2                                                  MAP05420
             N(3)=N3                                                    MAP05430
             DO 24 N4=1,2                                               MAP05440
                N(4)=N4                                                 MAP05450
                DO 25 N5=1,2                                            MAP05460
                   N(5)=N5                                              MAP05470
C                                                                       MAP05480
                   COEFF=1.                                             MAP05490
                   DO 30 M=1,5                                          MAP05500
                      COEFF=COEFF*T(N(M),M)                             MAP05510
   30              CONTINUE                                             MAP05520
C                                                                       MAP05530
                   XMAP = XMAP + COEFF*XHIT(N1,N2,N3,N4,N5)             MAP05540
                   YMAP = YMAP + COEFF*YHIT(N1,N2,N3,N4,N5)             MAP05550
                   ZMAP = ZMAP + COEFF*ZHIT(N1,N2,N3,N4,N5)             MAP05560
C                    UPDATE MAX-MIN RANGES                              MAP05570
                   IF (BFEXST(N1,N2,N3,N4,N5)) THEN                     MAP05580
                     IF (XHIT(N1,N2,N3,N4,N5).GT.XMAX) THEN             MAP05590
                       XMAX = XHIT(N1,N2,N3,N4,N5)                      MAP05600
                     ELSE                                               MAP05610
                       IF (XHIT(N1,N2,N3,N4,N5).LT.XMIN) THEN           MAP05620
                         XMIN = XHIT(N1,N2,N3,N4,N5)                    MAP05630
                       END IF                                           MAP05640
                     END IF                                             MAP05650
C                                                                       MAP05660
                     IF (YHIT(N1,N2,N3,N4,N5).GT.YMAX) THEN             MAP05670
                       YMAX = YHIT(N1,N2,N3,N4,N5)                      MAP05680
                     ELSE                                               MAP05690
                       IF (YHIT(N1,N2,N3,N4,N5).LT.YMIN) THEN           MAP05700
                         YMIN = YHIT(N1,N2,N3,N4,N5)                    MAP05710
```

```
                     END IF                                            MAP05720
                   END IF                                              MAP05730
C                                                                      MAP05740
                   IF (ZHIT(N1,N2,N3,N4,N5).GT.ZMAX) THEN              MAP05750
                     ZMAX = ZHIT(N1,N2,N3,N4,N5)                       MAP05760
                   ELSE                                                MAP05770
                     IF (ZHIT(N1,N2,N3,N4,N5).LT.ZMIN) THEN            MAP05780
                       ZMIN = ZHIT(N1,N2,N3,N4,N5)                     MAP05790
                     END IF                                            MAP05800
                   END IF                                              MAP05810
                 END IF                                                MAP05820
   25            CONTINUE                                              MAP05830
   24          CONTINUE                                                MAP05840
   23        CONTINUE                                                  MAP05850
   22      CONTINUE                                                    MAP05860
   21    CONTINUE                                                      MAP05870
      XRANGE = XMAX - XMIN                                             MAP05880
      YRANGE = YMAX - YMIN                                             MAP05890
      ZRANGE = ZMAX - ZMIN                                             MAP05900
      WRITE(6,*) 'MAPPING POINTS USED FOR INTERPOLATION WERE SPREAD:'  MAP05910
      WRITE(6,*) XRANGE,'RE ALONG THE XGSM DIRECTION'                  MAP05920
      WRITE(6,*) YRANGE,'RE ALONG THE YGSM DIRECTION'                  MAP05930
      WRITE(6,*) ZRANGE,'RE ALONG THE ZGSM DIRECTION'                  MAP05940
C                                                                      MAP05950
      RETURN                                                           MAP05960
      END                                                              MAP05970
C  ****************************** GETIP ****************************** MAP05980
      SUBROUTINE GETIP(BFPAR,TPRM,BFLIM,BFEXST)                        MAP05990
C                                                                      MAP06000
C     PURPOSE-    FIND THE MSM PARAMETER SETS WHICH LIE CLOSEST TO THE MAP06010
C                 INPUT PARAMETER SET AND LOAD THE B-FIELD MODEL       MAP06020
C                 INTERNAL PARAMETERS FROM THE APPROPRIATE MATRICES TO MAP06030
C                 BE PASSED ON FOR MAGNETIC FIELD LINE TRACING PURPOSES MAP06040
C     INPUT-      BFPAR: MSM INPUTS                                    MAP06050
C     OUTPUT-     TPRM: BFIELD MODEL INTERNAL INPUTS (14 FOR EACH)     MAP06060
C                 BFLIM:  VALUES FROM PARAMETER "GRID" WHICH BRACKET   MAP06070
C                         THE VALUES OF BFPAR                          MAP06080
C                 BFEXST: INDICATES IF A BFIELD MODEL AS REPRESENTED   MAP06090
C                         BY A MATRIX HAS BEEN LOADED.                 MAP06100
C     VARIABLES-  BFNDX: INDEX ASSIGNED TO BFLIM VALUES                MAP06120
C                 STNDPR,TILTPR,FINEDP,DSTPR,STCHPR: GRID VALUES OF THE MAP06130
C                    MSM INPUTS                                        MAP06140
C                 MSTND,MTILT,MINED,MDST,MSTCH: NUMBER OF GRID POINTS  MAP06150
C                    FOR EACH MSM INPUT                                MAP06160
C                 ICOUNT: NUMBER OF PARAMETER SETS LOADED.             MAP06170
C     SUBROUTINES-FNDBRK: FINDS THE SET OF B-FIELD MODELS WHICH        MAP06200
C                         BRACKET THE POINT OF INTEREST IN PARAMETER   MAP06210
C                         SPACE.                                       MAP06220
C                 LOADIP: LOADS INTERNAL B-FIELD PARAMETERS FROM MATRIX MAP06230
C                         FILES INTO VARIABLE TPRM.                    MAP06240
C       FUNCTIONS-MEXIST: INDICATES IF A BFIELD MODEL EXISTS           MAP06110
C                                                                      MAP06250
      PARAMETER (IMSTND=5,IMTILT=5,IMINED=16,IMDST=8,IMSTCH=2)         MAP06260
C                                                                      MAP06270
      LOGICAL MEXIST
      INTEGER BFNDX(2,5)                                               MAP06290
      LOGICAL*1 BFEXST(2,2,2,2,2)                                      MAP06300
C                                                                      MAP06310
      DIMENSION STNDPR(IMSTND),TILTPR(IMTILT),FINEDP(IMINED)           MAP06320
      DIMENSION DSTPR(IMDST),STCHPR(IMSTCH)                            MAP06330
      DIMENSION BFLIM(2,5),BFPAR(5)                                    MAP06340
      DIMENSION TPRM(14,2,2,2,2,2)                                     MAP06350
      DATA MSTND/IMSTND/                                               MAP06360
      DATA MTILT/IMTILT/                                               MAP06370
```

```
      DATA MINED/IMINED/                                                MAP06380
      DATA MDST/IMDST/                                                  MAP06390
      DATA MSTCH/IMSTCH/                                                MAP06400
C                                                                       MAP06410
      DATA STNDPR/6.0,8.0,10.0,12.0,14.0/                               MAP06420
      DATA TILTPR/-35.0,-17.5,0.0,17.5,35.0/                            MAP06430
      DATA FINEDP/49.47, 53.20, 55.89, 57.96, 59.60,                    MAP06440
     1            60.95, 62.06, 63.02, 63.99, 65.01,                    MAP06450
     2            65.91, 66.72, 67.45, 68.12, 68.73,                    MAP06460
     3            69.30/                                                MAP06470
      DATA DSTPR/-400.0, -300.0, -200.0, -150.0,                        MAP06480
     1           -100.0,  -50.0,    0.0,   50.0/                        MAP06490
      DATA STCHPR/0.0, 1.0/                                             MAP06500
C
      EXTERNAL MEXIST
C                                                                       MAP06510
      CALL FNDBRK(BFPAR(1),STNDPR,MSTND,BFNDX(1,1),BFNDX(2,1))          MAP06540
      CALL FNDBRK(BFPAR(2),TILTPR,MTILT,BFNDX(1,2),BFNDX(2,2))          MAP06550
      CALL FNDBRK(BFPAR(3),FINEDP,MINED,BFNDX(1,3),BFNDX(2,3))          MAP06560
      CALL FNDBRK(BFPAR(4),DSTPR, MDST, BFNDX(1,4),BFNDX(2,4))          MAP06570
      CALL FNDBRK(BFPAR(5),STCHPR,MSTCH,BFNDX(1,5),BFNDX(2,5))          MAP06580
C                                                                       MAP06590
      DO 5, I=1,2                                                       MAP06600
         BFLIM(I,1) = STNDPR(BFNDX(I,1))                                MAP06610
         BFLIM(I,2) = TILTPR(BFNDX(I,2))                                MAP06620
         BFLIM(I,3) = FINEDP(BFNDX(I,3))                                MAP06630
         BFLIM(I,4) = DSTPR(BFNDX(I,4))                                 MAP06640
         BFLIM(I,5) = STCHPR(BFNDX(I,5))                                MAP06650
 5    CONTINUE                                                          MAP06660
C                                                                       MAP06670
      ICOUNT = 0                                                        MAP06680
      DO 10, I=1,2                                                      MAP06690
        DO 20, J=1,2                                                    MAP06700
          DO 30, K=1,2                                                  MAP06710
            DO 40, L=1,2                                                MAP06720
              DO 50, M=1,2                                              MAP06730
                     IF (MEXIST(BFNDX(I,1),BFNDX(J,2),BFNDX(K,3),       MAP06740
     1                          BFNDX(L,4),BFNDX(M,5))) THEN            MAP06750
                        CALL LOADIP(I,J,K,L,M,BFNDX,TPRM)               MAP06760
                        BFEXST(I,J,K,L,M) = .TRUE.                      MAP06770
                        ICOUNT = ICOUNT + 1                             MAP06780
                     ELSE                                               MAP06790
                        BFEXST(I,J,K,L,M) = .FALSE.                     MAP06800
                     ENDIF                                              MAP06810
 50           CONTINUE                                                  MAP06820
 40         CONTINUE                                                    MAP06830
 30       CONTINUE                                                      MAP06840
 20     CONTINUE                                                        MAP06850
 10   CONTINUE                                                          MAP06860
      IF (ICOUNT.LT.8) THEN                                             MAP06870
        WRITE(6,1000) ICOUNT                                            MAP06880
 1000   FORMAT(1X,'ONLY',I3,' MODELS AVAILABLE (NEED 8 MINIMUM)')       MAP06890
        WRITE(6,*) 'B-FIELD TRACING INTERPOLATION PROCEDURE INVALIDATED'MAP06900
        WRITE(6,*) 'EXITED FROM SUBROUTINE GETIP'                       MAP06910
        RETURN                                                          MAP06920
      END IF                                                            MAP06930
C                                                                       MAP06940
      RETURN                                                            MAP06950
      END                                                               MAP06960
C                                                                       MAP06970
C   ******************************* LOADIP *****************************MAP06980
      SUBROUTINE LOADIP(I,J,K,L,M,BFNDX,TPRM)                           MAP06990
C                                                                       MAP07000
C   PURPOSE-                                                            MAP07010
```

```
C    THIS SUBROUTINE LOADS INTERNAL MODEL PARAMETERS FROM OFFLINE
C    B-SUPERMATRIX INTO THE TPRM MATRIX.  A CHECK IS MADE TO
C    VERIFY THAT THE CORRECT PARAMETERS HAVE BEEN RETRIEVED.  IF THIS
C    CHECK FAILS THE PROGRAM WILL STOP. THIS SUBROUTINE HAS THE SAME
C    BASIC STRUCTURE (AND READS FROM THE SAME DATA FILES) AS SUBROUTINE
C    LOADBM. ONE CALL PER MODEL.
C   INPUT-    I,J,K,L,M: INDICES FOR MATRIX BFNDX
C             BFNDX: INDEX NUMBERS OF PARAMETERS USED TO IDENTIFY
C                    INDIVIDUAL MATRICES.
C   OUTPUT-   TPRM: INTERNAL B-FIELD PARAMETERS (14) FOR MAXIMUM OF 32
C                   DIFFERENT MODELS (2X2X2X2X2)
C   VARIABLES-IDF1-IDF6: MATRIX IDENTIFIERS
C
      INTEGER BFNDX(2,5)
      CHARACTER*32 FILNAM
C
      DIMENSION TPRM(14,2,2,2,2,2)
C
         WRITE (FILNAM,1000) BFNDX(J,2), BFNDX(I,1), BFNDX(L,4),
     1                       BFNDX(K,3),(3-BFNDX(M,5))
 1000    FORMAT ('BFIELD:BO',I1,I1,I1,I2.2,I1,'.DAT')
C
         OPEN (UNIT=99,FILE=FILNAM,ERR=199,FORM='FORMATTED',
     1         ACCESS='SEQUENTIAL')
C
         READ (99,800) IDF1,IDF2,IDF3,IDF4,IDF5,IDF6,TPRM( 1,I,J,K,L,M),
     1        TPRM( 2,I,J,K,L,M),TPRM( 3,I,J,K,L,M),TPRM( 4,I,J,K,L,M),
     2        TPRM( 5,I,J,K,L,M),TPRM( 6,I,J,K,L,M),TPRM( 7,I,J,K,L,M),
     3        TPRM( 8,I,J,K,L,M),TPRM( 9,I,J,K,L,M),TPRM(10,I,J,K,L,M),
     4        TPRM(11,I,J,K,L,M),TPRM(12,I,J,K,L,M),TPRM(13,I,J,K,L,M),
     5        TPRM(14,I,J,K,L,M)
  800 FORMAT(3I1,I2.2,I1,I3,14F8.2)
C
         IF ((BFNDX(I,1).NE.IDF2).OR.(BFNDX(J,2).NE.IDF1).OR.
     1      (BFNDX(K,3).NE.IDF4).OR.(BFNDX(L,4).NE.IDF3).OR.
     2      ((3-BFNDX(M,5)).NE.IDF5).OR.(IDF6.NE.201)) THEN
            WRITE (6,*) 'FILE ',FILNAM,' IS INCORRECT.'
            STOP 'INCORRECT B-FIELD. STOPPING IN LOADIP.'
         ENDIF
C
         CLOSE (99)
         RETURN
C
 199     CONTINUE
         WRITE(6,*) 'STOPPING IN LOADIP. OPEN FAILED ON ',FILNAM
         STOP 'STOPPING IN LOADBM.  OPEN FAILED ON B-MATRIX.'
      END
C********************************* BMNMAP
      SUBROUTINE BMNMAP(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1                  COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,X,Y,Z,
     2                  IOPEN,XHIT,YHIT,ZHIT)
C
C     PURPOSE-    MAGNETIC FIELD LINE MAPPING TO THE MINIMUM |B| SURFACE
C                 OR THE MAGNETOPAUSE BOUNDARY, OPEN/CLOSED DETERMINED
C     INPUT-      FIRST 15: BFIELD INPUT
C                 RADIUS: MAGNETOTAIL RADIUS FOR DEFINING BOUNDARY
C                 X,Y,Z,: GSM STARTING POINT OF TRACE
C     OUTPUT-     IOPEN: DESIGNATES OPEN FIELD LINE   = 1
C                                   CLOSED FIELD LINE = 0
C                 XHIT,YHIT,ZHIT: LAST TRACE POINT. EITHER AT MSM
C                      SURFACE WHEN FIELD LINE IS CLOSED OR AT A POINT
C                      ON THE DEFINED MAGNETOPAUSE FOR OPEN LINES.
C     VARIABLE-   ER,EP,H:ERROR,DIRECTION, AND INITIAL TRACE STEP SIZE
C                 DX1,DY1,DZ1,BB: VECTOR INFORMATION FOR SUBROUT. TRAC
```

```
C                BDIST: = RADIUS - STAND-OFF DISTANCE
C                DIST: DISTANCE FROM X-AXIS
C                XP: = XN + BDIST (DISTANCE FROM CYLINDER-SPHERE
C                    INTERFACE IN X DIRECTION)
C                DDAY: DISTANCE FROM CYLINDER-SPHERE INTERFACE POINT
C                      WHERE YGSM = ZGSM = 0
C                XA,YA,ZA: STARTING POINT FOR TRACING STEP
C                XN,YN,ZN: FINISH POINT FOR TRACING STEP
C                BMAGA,BMAGN: MAGNETIC FIELD MAGNITUDE AT POINTS
C                          (XA,YA,ZA) AND (XN,YN,ZN) RESPECTIVELY.
C                          USED WHEN TRACE DIRECTION IS BEING DECIDED.
C                BX,BY,BZ: MAGNETIC FIELD STRENGTH (NT) DURING TRACE.
C                BSIZE: MAGNETIC FIELD MAGNITUDE
C                BMIN: CURRENT SMALLEST B-FIELD MAGNITUDE
C     SUBROUTINE- TRAC: FINDS SEQUENTIAL STEPS IN VECTOR TRACE
C                BFIELD: MAGNETIC FIELD MODEL POINT CALCULATIONS
C
      COMMON/BSTEP/DX1,DY1,DZ1,BB,H,EP,ER
C
C          SET TRACING ERROR LIMITS,INITIAL STEP SIZE, AND DIRECTION
      ER =  0.25E-02
      H  =  0.25
      EP = -1.0
C       INITIALIZE QUANTITIES
      IOPEN = 1
      XHIT = 999.
      YHIT = 999.
      ZHIT = 999.
      XA = X
      YA = Y
      ZA = Z
      DX1=0.
      DY1=0.
      DZ1=0.
      BDIST = RADIUS - STAND
C       SET BMIN START VALUE, ASSUME START POINT |B| VALUE
      CALL BFIELD(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)
      BMIN = SQRT(BX**2 + BY**2 + BZ**2)
C       CHECK THAT DIRECTION OF TRACE IS TOWARDS MINIMUM
      CALL TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,
     1          DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)
C       GET FIELD STRENGTH AT STARTING POINT
      CALL BFIELD(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXA,BYA,BZA)
C       GET FIELD STRENGTH AT FIRST TRACER STEP POINT
      CALL BFIELD(ISWIT,XN,YN,ZN,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXN,BYN,BZN)
      BMAGA = SQRT(BXA**2 + BYA**2 + BZA**2)
      BMAGN = SQRT(BXN**2 + BYN**2 + BZN**2)
      IF (BMAGN.GT.BMAGA) THEN
C         MOVING AWAY FROM MINIMUM, CHANGE DIRECTION
        EP = -EP
        CALL TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,
     1            DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)
C         GET FIELD STRENGTH AT FIRST TRACER STEP POINT IN NEW DIR.
        CALL BFIELD(ISWIT,XN,YN,ZN,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1              COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXN,BYN,BZN)
        BMAGN = SQRT(BXN**2 + BYN**2 + BZN**2)
        IF (BMAGN.GT.BMAGA) THEN
C         FIRST POINT IN EITHER DIRECTION HAS STRONGER FIELD
C         MUST HAVE STARTED ON THE MINIMUM |B| SURFACE
          IOPEN = 0
          XHIT = XA
```

```
            YHIT = YA
            ZHIT = ZA
            GOTO 20
          END IF
        END IF
 30     CONTINUE
C         START FULL TRACE OF A SINGLE FIELD-LINE
        CALL TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,
     1            DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)
        DIST = SQRT(YN**2 + ZN**2)
        XP = XN + BDIST
        DDAY = SQRT(XP**2 + YN**2 + ZN**2)
        IF ((XN.LT.(-50.)).OR.(DIST.GT.RADIUS).OR.
     1                      ((XP.GE.0.).AND.(DDAY.GT.RADIUS))) THEN
C           TRACE HIT MAGNETOPAUSE OR FAR TAIL LIMIT (OPEN FIELD LINE)
          IOPEN = 1
          XHIT = XN
          YHIT = YN
          ZHIT = ZN
          WRITE(6,*) '**************************************************'
          WRITE(6,*) 'OPEN FIELD-LINE HIT AT POINT: XGSM    YGSM    ZGSM'
          WRITE(6,*) XHIT,YHIT,ZHIT
          GOTO 20
        END IF
C
        CALL BFIELD(ISWIT,XN,YN,ZN,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1             COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)
        BSIZE = SQRT(BX**2 + BY**2 + BZ**2)
        IF (BSIZE.LE.BMIN) THEN
C           CLOSER TO MINIMUM IN |B|
          BMIN = BSIZE
          XHIT = XN
          YHIT = YN
          ZHIT = ZN
          IOPEN = 0
        ELSE
C           PAST MINIMUM |B|, DONE
          GOTO 20
        END IF
C
        XA = XN
        YA = YN
        ZA = ZN
        GOTO 30
C
 20     CONTINUE
        RETURN
        END
C*************************** TRAC
        SUBROUTINE TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,
     1                 DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)
C
C       PURPOSE-     FINDS SUCCESSIVE STEP LOCATION IN A TRACE OF A VECTOR
C                    FIELD, E.G., USED TO MAP MAGNETIC FIELD LINES.
C       INPUT-       FIRST 15 ARE BFIELD MODEL INPUTS (SUBROUTINE BFIELD)
C                    XA,YA,ZA: GSM LOCATION OF START POINT OF VECTOR TRACE.
C       OUTPUT-      RADIUS: MAGNETOTAIL RADIUS (RE)
C                    XN,YN,ZN: GSM LOCATION OF NEXT STEP IN VECTOR TRACE.
C       VARIABLES-   H:  STEP SIZE
C                    EP: DIRECTION OF TRACE (+1. IS PARALLEL TO FIELD
C                        VECTOR AND -1. IS ANTIPARALLEL TO SAME)
C                    ER: ALLOWABLE DISTANCE ERROR LIMIT (RE)
C       SUBROUTINES-FUNC: PROVIDES VECTOR INFORMATION ABOUT THE BFIELD
C
```

```
      COMMON/BSTEP/DX1,DY1,DZ1,BB,H,EP,ER                               MAP08940
      REAL DX1,DY1,DZ1,BB,H,EP,ER,XA,YA,ZA,XN,YN,ZN                     MAP08950
C                                                                       MAP08960
      CALL FUNC(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,     MAP08970
     1          COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX1,DY1,DZ1,BF1) MAP08980
 1    CONTINUE                                                          MAP08990
      H1=H*EP                                                           MAP09000
      H2=0.5*H1                                                         MAP09010
C                                                                       MAP09020
      XN1=XA+H1*DX1                                                     MAP09030
      YN1=YA+H1*DY1                                                     MAP09040
      ZN1=ZA+H1*DZ1                                                     MAP09050
      XN2=XA+H2*DX1                                                     MAP09060
      YN2=YA+H2*DY1                                                     MAP09070
      ZN2=ZA+H2*DZ1                                                     MAP09080
C                                                                       MAP09090
      CALL FUNC(ISWIT,XN2,YN2,ZN2,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,  MAP09100
     1          COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX2,DY2,DZ2,BF2) MAP09110
      XN3=XN2+H2*DX2                                                    MAP09120
      YN3=YN2+H2*DY2                                                    MAP09130
      ZN3=ZN2+H2*DZ2                                                    MAP09140
C                                                                       MAP09150
      DELX=XN1-XN3                                                      MAP09160
      DELY=YN1-YN3                                                      MAP09170
      DELZ=ZN1-ZN3                                                      MAP09180
      RTES1=DELX*DELX                                                   MAP09190
      RTES2=DELY*DELY                                                   MAP09200
      RTES3=DELZ*DELZ                                                   MAP09210
      RTES=SQRT(RTES1+RTES2+RTES3)                                      MAP09220
C                                                                       MAP09230
      IF(RTES-ER) 4,4,2                                                 MAP09240
 4    CONTINUE                                                          MAP09250
C                                                                       MAP09260
      XN=XN3-DELX                                                       MAP09270
      YN=YN3-DELY                                                       MAP09280
      ZN=ZN3-DELZ                                                       MAP09290
      BB=BF2                                                            MAP09300
      IF(RTES-(0.25*ER)) 7,6,6                                          MAP09310
 7    H=2.*H                                                            MAP09320
C  ***LIMIT MAXIMUM STEP SIZE***                                        MAP09330
      IF (H.GT.(5.0)) THEN                                              MAP09340
        H = 5.0                                                         MAP09350
      END IF                                                            MAP09360
 6    GOTO 5                                                            MAP09370
C                                                                       MAP09380
 2    CONTINUE                                                          MAP09390
      H=0.5*H                                                           MAP09400
      GOTO 1                                                            MAP09410
C                                                                       MAP09420
 5    CONTINUE                                                          MAP09430
      RETURN                                                            MAP09440
      END                                                               MAP09450
C*************************** FUNC                                      MAP09460
     SUBROUTINE FUNC(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,   MAP09470
     1               COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX,DYY,DZ,B)MAP09480
C                                                                       MAP09490
C    PURPOSE-   PROVIDE VECTOR INFORMATION ABOUT A FUNCTION (BFIELD)    MAP09500
C    INPUT-     FIRST 18: BFIELD INPUTS                                 MAP09510
C    OUTPUT-    DX,DYY,DZ,B: VECTOR COMPONENT INFORMATION               MAP09520
C    SUBROUTINE-BFIELD: MAGNETIC FIELD MODEL                            MAP09530
C                                                                       MAP09540
     BX = 0.                                                            MAP09550
     BY = 0.                                                            MAP09560
     BZ = 0.                                                            MAP09570
```

```
      CALL BFIELD(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
     1            COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)
C
      B = SQRT(BX**2 + BY**2 + BZ**2)
      DX = BX/B
      DYY = BY/B
      DZ = BZ/B
C
      RETURN
      END
      SUBROUTINE CHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,IFLAG)
C
C  THIS SUBROUTINE CHECKS THAT THE INTERACTIVE INPUT IS WITHIN
C   REASONABLE LIMITS.
C
      IFLAG=0
      IF((IYEAR.LT.0).OR.(IYEAR.GT.99))THEN
       WRITE(6,*)' INCORRECT INPUT FOR YEAR-MUST BE BETWEEN 0 AND 99'
       IFLAG= 1
       RETURN
      ENDIF
      IF((IDAY.LT.1).OR.(IDAY.GT.366))THEN
       WRITE(6,*)' INCORRECT INPUT FOR DAY-MUST BE BETWEEN 1 AND 366'
       IFLAG= 1
       RETURN
      ENDIF
      IF((IHOUR.LT.0).OR.(IHOUR.GT.24))THEN
       WRITE(6,*)' INCORRECT INPUT FOR HOUR-MUST BE BETWEEN 1 AND 24'
       IFLAG=1
       RETURN
      ENDIF
      IF((MINUTE.LT.0).OR.(MINUTE.GT.60))THEN
       WRITE(6,*)' INCORRECT INPUT FOR MINUTE-MUST BE BETWEEN 0 AND 60'
       IFLAG= 1
       RETURN
      ENDIF
      IF((ISCNDS.LT.0).OR.(ISCNDS.GT.60))THEN
       WRITE(6,*)' INCORRECT INPUT FOR SECONDS-MUST BE BETWEEN 0 AND 60'
       IFLAG= 1
       RETURN
      ENDIF
      RETURN
      END
      SUBROUTINE TCHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,AUGPAR,AGPAR1,
     1   NAUGEL,IFLAG,ISTART)
C
C  THIS SUBROUTINE CHECKS TO FIND THE RECORD NUMBERS OF THE DATA
C   FILES WHICH SPAN THE REQUESTED TIME.
C
      DIMENSION AUGPAR(NAUGEL),AGPAR1(NAUGEL),ISTART(3),ITIME(3)
      IFLAG=0
      ITIME(1)=IYEAR
      ITIME(2)=IDAY
      ITIME(3)=IHOUR*3600 + MINUTE*60 + ISCNDS
      TIMCHK=TCONV3(ITIME,ISTART)
      ITIME(1)=AUGPAR(1)
      ITIME(2)=AUGPAR(2)
      ITIME(3)=AUGPAR(3)*3600 + AUGPAR(4)*60 + AUGPAR(5)
      TIMAUG=TCONV3(ITIME,ISTART)
      ITIME(1)=AGPAR1(1)
      ITIME(2)=AGPAR1(2)
      ITIME(3)=AGPAR1(3)*3600 + AGPAR1(4)*60 + AGPAR1(5)
      TIMAG1=TCONV3(ITIME,ISTART)
      IF((TIMCHK.GE.TIMAUG).AND.(TIMCHK.LT.TIMAG1))THEN
```

```
         IFLAG=0
        ELSE
         IFLAG=1
        ENDIF
        RETURN
        END
C    ****************************** TCONV3 ******************************
C
        REAL FUNCTION TCONV3(ITIME,ISTART)
C
C    THIS FUNCTION TAKES A START YEAR, START DAY, A CURRENT YEAR,
C    A CURRENT DAY, AND A CURRENT SECOND..  IT PRODUCES
C    A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
C    THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.
C
        PARAMETER (IYR=1,IDY=2,ISEC=3)
        DIMENSION ITIME(3),ISTART(3)
        INTRINSIC REAL
C
           IF (ISTART(IYR).EQ.ITIME(IYR)) THEN
              TEMP = REAL(ITIME(IDY) - ISTART(IDY))*86400.0 +
     1               REAL(ITIME(ISEC))
           ELSEIF (MOD(ISTART(IYR),4).EQ.0) THEN
              TEMP = REAL(ITIME(IDY) + 366 - ISTART(IDY))*86400.0 +
     1               REAL(ITIME(ISEC))
           ELSE
              TEMP = REAL(ITIME(IDY) + 365 - ISTART(IDY))*86400.0 +
     1               REAL(ITIME(ISEC))
           ENDIF
C
           TCONV3 = TEMP
C
           RETURN
        END
        SUBROUTINE LOCATE(RVAL,PVAL,LATDIM,LTDIM,R,P,BI,BJ)
C
C   PURPOSE:  FIND GRID LOCATION (BI,BJ) OF PHYSICAL LOCATIONS GIVEN BY
C             (RVAL,PVAL,LL)
C
C   VERSION 1.00                       DATE: 01.22.90
C
C
C   INPUT:
C         RVAL      RADIAL DISTANCE (RE)
C         PVAL      HOUR ANGLE MEASURED EASTWARD FROM NOON (RADIAN)
C         LATDIM    NUMBER OF LATITUDINAL GRID PTS
C         LTDIM     MAX NUMBER OF LOCAL TIME GRID PTS (INCL WRAPAROUND)
C         R         RADIAL DISTANCE ARRAY (RE)
C         P         HOUR ANGLE ARRAY (RADIANS)
C
C   OUTPUT:
C         BI        NON-INTEGER I LOCATION OF (RVAL,XLT)
C         BJ        NON-INTEGER J LOCATION OF (RVAL,XLT)
C
C   PROGRAMMER: R.W. SPIRO
C
C
C
C
C
C
        DIMENSION R(LATDIM,LTDIM),P(LATDIM,LTDIM)
C
C
```

```
      PI=ATAN2(0.,-1.)
C
C
C
C
C
C
C
C  SEARCH GRID AND TRY TO LOCALIZE (RVAL,PVAL) WITHIN A GRID SQUARE
      DO 30 J=2,LTDIM-2
         JJ=J
C
         DO 40 I=2,LATDIM
            II=I
C
C  GET PVAL,P(I,J), AND P(I,J+1) IN SAME MODULUS
            CALL PFIX(PVAL,P(I,J),P(I,J+1),P4,P5)
C           WRITE(6,*) I,J,RVAL,R(I-1,J,LL),R(I,J,LL),PVAL,P4,P5
C
            IF(RVAL.LE.R(I-1,J)
     2         .AND.
     3             RVAL.GT.R(I,J)
     4         .AND.
     5             PVAL.GE.P4
     6         .AND.
     7             PVAL.LT.P5)
     8      THEN
               GO TO 50
            END IF
C
   40   CONTINUE
   30 CONTINUE
C
      WRITE(6,*) 'STOPPING IN LOCATE'
      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
      WRITE(6,*) 'RVAL= ',RVAL,'  PVAL= ',PVAL
      STOP
C
C
   50 CONTINUE
C
      DO 45 IK=2,LATDIM
         IIK=IK
         IF(RVAL.LE.R(IK-1,J+1).AND.RVAL.GT.R(IK,J+1)) GO TO 60
   45 CONTINUE
C
      WRITE(6,*) 'STOPPING IN EFLOC'
      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
      WRITE(6,*) 'RVAL= ',RVAL,'  PVAL= ',PVAL
      STOP
C
   60 CONTINUE
C
C
C  COMPUTE COEFFICIENTS FOR INTERPOLATION
      F1=(RVAL-R(II-1,JJ))/(R(II,JJ)-R(II-1,JJ))+FLOAT(II-1)
      F2=(RVAL-R(IIK-1,JJ+1))/(R(IIK,JJ+1)-R(IIK-1,JJ+1))+
     2    FLOAT(IIK-1)
C
      CALL PFIX(XLT,P(II,JJ),P(IIK,JJ+1),P4,P5)
C
      BJ=(PVAL-P4)/(P5-P4)+FLOAT(JJ)
C
      IF(BJ.LT.1.) BJ=BJ+FLOAT(LTDIM-3)
```

```
      IF(BJ.GT.FLOAT(LTDIM)) BJ=BJ-FLOAT(LTDIM-3)
C
C
      BI=F1+(BJ-FLOAT(JJ))*(F2-F1)/(FLOAT(JJ+1)-FLOAT(JJ))
C
C
      RETURN
      END
C
      SUBROUTINE PFIX(P1,P2,P3,P4,P5)
C
C  PURPOSE:  SUBROUTINE TO ADJUST MODULUS OF P2 AND P3 TO MATCH P1.
C          ON OUTPUT P4 CORRESPONDS TO P2, P5 TO P3
C
C  VERSION 1.00                        DATE: 09.04.89
C
C  PROGRAMMER: R.W. SPIRO
C
C
      PI=ATAN2(0.,-1.)
C
      P4=P2
      P5=P3
C
      IF(ABS(P2-P3).GT.PI) THEN
         IF(P2.LT.P3) THEN
            P4=P2+2.*PI
         ELSE
            P5=P3+2.*PI
         END IF
C
         IF(P1.LT.PI) THEN
            P4=P4-2.*PI
            P5=P5-2.*PI
         END IF
C
      END IF
C
C
      RETURN
      END
      FUNCTION G3NTRP(A,IMAX,JMAX,KMAX,BI,BJ,BK)
C
C  VERSION 1.00                            DATE: 01.11.88
C          1.01A                                 02.02.89

C
C  PURPOSE: FUNCTION SUBPROGRAM TO PERFORM A GENERAL 3-D LINEAR
C           INTERPOLATION OF ARRAY A(I,J,K) AT PT(BV(1),BV(2),BV(3))
C
C  INPUT:
C      A                     3-D ARRAY TO BE INTERPOLATED
C                            INTERPOLATE A
C
C
C
         COMMON /LUNIT/LUERR,LUPPT,LUCORD,LUPRNT,LUIDAT,LUHDYE,
     1      LSPARE,LUETAB,LUEBEG,LUPLEC,LUPION,LUEAVG,LUFLSM,LUETA,
     2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
     3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION A(IMAX,JMAX,KMAX)
      DIMENSION NDX(3),NDIM(3),BV(3),COEF(3,2)
C
C  PREPARE INDICES FOR INTERPOLATION
```

```
C                                                                       MAP12140
      NDIM(1)=IMAX                                                      MAP12150
      NDIM(2)=JMAX                                                      MAP12160
      NDIM(3)=KMAX                                                      MAP12170
      BV(1)=BI                                                          MAP12180
      BV(2)=BJ                                                          MAP12190
      BV(3)=BK                                                          MAP12200
      DO 10 L=1,3                                                       MAP12210
         NDX(L)=BV(L)                                                   MAP12220
         IF(NDX(L).LT.1) NDX(L)=1                                       MAP12230
         IF(NDX(L).GT.NDIM(L)-1) NDX(L)=NDIM(L)-1                       MAP12240
         IF(NDX(L).LE.0) NDX(L)=1                                       MAP12250
C                                                                       MAP12260
         FNDX=REAL(NDX(L))                                              MAP12270
         COEF(L,1)=1.-BV(L)+FNDX                                        MAP12280
         COEF(L,2)=BV(L)-FNDX                                           MAP12290
   10 CONTINUE                                                          MAP12300
C                                                                       MAP12310
C                                                                       MAP12320
      G3NTRP=0.                                                         MAP12330
      KSTOP = MIN(KMAX,2)                                               MAP12340
      JSTOP = MIN(JMAX,2)                                               MAP12350
      DO 20 I=1,2                                                       MAP12360
      DO 20 J=1,JSTOP                                                   MAP12370
      DO 20 K=1,KSTOP                                                   MAP12380
         G3NTRP=G3NTRP+                                                 MAP12390
     1          COEF(1,I)*COEF(2,J)*COEF(3,K)*A(NDX(1)+I-1,NDX(2)+J-1,  MAP12400
     2                                          NDX(3)+K-1)             MAP12410
   20 CONTINUE                                                          MAP12420
      RETURN                                                            MAP12430
      END                                                               MAP12440
C                                                                       MAP12450
C                                                                       MAP12460
      SUBROUTINE READ3D(LUN,FILNAM,LREC,IDIM,JDIM,KDIM,ITMDIM,ID,RID,   MAP12700
     2        CHID,ARRAY)                                               MAP12710
C                                                                       MAP12720
C     PROGRAMMER: BOB SPIRO                DATE: 6/26/88                MAP12730
C                                                                       MAP12740
C     PURPOSE:  THIS SUBROUTINE READS A LOGICAL RECORD FROM THE         MAP12750
C     STANDARD MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.         MAP12760
C                                                                       MAP12770
C                                                                       MAP12780
      CHARACTER CHID*80, FILNAM*6                                       MAP12790
C                                                                       MAP12800
         DIMENSION ARRAY(IDIM,JDIM,KDIM)                                MAP12810
         DIMENSION ID(20),RID(20)                                       MAP12820
C                                                                       MAP12830
C        OPEN THE FILE                                                  MAP12840
C                                                                       MAP12850
         IRECL = IDIM*JDIM + 2*80                                       MAP12860
         OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C                                                                       MAP12880
C        READ THE ARRAY                                                 MAP12890
C                                                                       MAP12900
         DO 10, K=1, KDIM                                               MAP12910
C                                                                       MAP12920
C           CALCULATE THE RECORD NUMBER                                 MAP12930
C                                                                       MAP12940
            IREC = (LREC-1)*KDIM + K                                    MAP12950
            READ (LUN,REC=IREC) ID,RID,CHID,                            MAP12960
     2            ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)                    MAP12970
 10      CONTINUE                                                       MAP12980
C                                                                       MAP12990
         CLOSE(LUN)                                                     MAP13000
```

```
C                                                                       MAP13010
         RETURN                                                         MAP13020
C                                                                       MAP13030
      END                                                               MAP13040
C                                                                       MAP13050
C   ****************************** FNDBRK *****************************MAP13060
C                                                                       MAP13070
      SUBROUTINE FNDBRK(PARVAL,PVALS,IPDIM,MIN,MAX)                     MAP13080
C                                                                       MAP13090
C   THIS SUBROUTINE FINDS THE INDICES OF THE VALUES IN ARRAY            MAP13100
C   PVALS THAT BRACKET PARVAL.                                          MAP13110
C                                                                       MAP13120
C   BRYAN BALES                   9/12/89                               MAP13130
C                                                                       MAP13140
      DIMENSION PVALS(IPDIM)                                            MAP13150
C                                                                       MAP13160
         MIN = 0                                                        MAP13170
C                                                                       MAP13180
 10      CONTINUE                                                       MAP13190
            MIN = MIN + 1                                               MAP13200
            MAX = MIN + 1                                               MAP13210
C        IF (PARVAL.LE.PVALS(MIN)) THEN                                 MAP13220
C           MAX = MIN                                                   MAP13230
C        ELSEIF (PARVAL.GT.PVALS(MAX).AND.MAX.EQ.IPDIM) THEN            MAP13240
C           MIN = MAX                                                   MAP13250
         IF (PARVAL.GT.PVALS(MAX).AND.MAX.LT.IPDIM) THEN                MAP13260
            GO TO 10                                                    MAP13270
         ENDIF                                                          MAP13280
C                                                                       MAP13290
         RETURN                                                         MAP13300
      END                                                               MAP13310
C                                                                       MAP13320
      LOGICAL FUNCTION MEXIST(MSTND,MTILT,MINED,MDST,MSTCH)
C
C       THIS FUNCTION CHECKS TO SEE IF A PARTICULAR B-FIELD IS
C       AVAILABLE.
C
         CHARACTER*32 FILNAM
         LOGICAL TEST

         WRITE (FILNAM,1000) MTILT,MSTND,MDST,MINED,(3-MSTCH)
 1000    FORMAT ('BFIELD:BO',I1,I1,I1,I2.2,I1,'.DAT')

         INQUIRE (FILE=FILNAM,EXIST=TEST)
         MEXIST = TEST
C
         RETURN
      END
C
      SUBROUTINE RDVEC(NOUNIT,IRDIN,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
C
C  THIS SUBROUTINE READS THE HEADER VECTOR FOR THE ETA FILES.  THE
C   HEADER CONTAINDS THE ALAM, THRESHOLD ENERGY, AND ERROR INFORMATION
C   FOR THIS RUN.
C
      DIMENSION ALAM(IEDIM),THRSH(IEDIM),ERSHFT(IEDIM),IFLAV(IEDIM)     MAP14560
C                                                                       MAP14570
C   OPEN THE FILE                                                       MAP14580
C                                                                       MAP14590
      IRECL = 4*iedim + 1                                               MAP14600
      OPEN(UNIT=NOUNIT,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C                                                                       MAP14620
C  READ VALUES                                                          MAP14630
      READ (NOUNIT,REC=IRDIN) IEMAX,ALAM,THRSH,ERSHFT,IFLAV
```

```
C
      CLOSE (NOUNIT)
      RETURN
      END
      SUBROUTINE SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)
C
C THIS SUBROUTINE READS THE NUMBER OF ENERGY CHANNELS AND THEIR
C  ENERGIES AT GEOSYNCHRONOUS ORBIT AND CALCULATES THE ENERGY
C  INVARIANTS NEEDED TO RUN THE PROGRAM.
C THE INPUT ENERGY IS IN EV AND MUST BE IN ASCENDING ORDER SO THAT THE
C  ALMDEL VARIABLES CAN BE CORRECTLY CALCULATED.
C
C
      PARAMETER(VM6=7.0)
C
      DIMENSION ALAM(IEDIM)
      DIMENSION IFLAV(IEDIM)
      DIMENSION ALMDEL(IEDIM)
      DIMENSION NUM(3)
C
      DO 5 N=1,3
         NUM(N)=0
    5 CONTINUE
C
C  COMPUTE ALMDEL
      DO 20 IE=1,IEMAX
       ISP=IFLAV(IE)
       IF(NUM(ISP).GE.2) THEN
         IF(IE.EQ.1.OR.IFLAV(IE-1).NE.ISP) THEN
            ALMMIN=0.
         ELSE
            ALMMIN=(ALAM(IE-1)+ALAM(IE))/2.
         END IF
C
         IF(IE.EQ.IEMAX.OR.IFLAV(IE+1).NE.ISP) THEN
            ALMMAX=1.5*ALAM(IE)-.5*ALAM(IE-1)
         ELSE
            ALMMAX=(ALAM(IE)+ALAM(IE+1))/2.
         END IF
C
         ALMDEL(IE)=ABS(ALMMAX)-ABS(ALMMIN)
C
       END IF
   20 CONTINUE
C
C
       RETURN
       END
      SUBROUTINE BFIELD(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,
     1           D,COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,
     2           BXTOT,BYTOT,BZTOT)
C
C
C    PURPOSE-   CALLS ALL OF THE MAGNETIC FIELD MODEL ROUTINES FOR A
C               GIVEN POINT AND SUMS UP THE MAGNETIC FIELD VECTORS (NO
C               CURRENT) CONTRIBUTIONS. THE THREE ROUTINES ARE BVOIGT,
C               TAILC, AND RINGC, REPRESENTING A SHIELDED VACUUM DIPOLE,
C               A TAIL CURRENT SYSTEM, AND A RING CURRENT SYSTEM,RESP.
C               MUST CALL TWICE, BEGINNING WITH ISWIT=0, TO CALCULATE
C               AND COLLECT CONSTANTS. THE THIRD AND HIGHER CALLS GIVE
C               RESULTS FOR SINGLE POINTS X,Y,Z IN GSM COORDINATES.
C               THE COUNTER ISWIT IS ADJUSTED INTERNALLY SO IT NEED ONLY
C               BE SPECIFIED (=0) FOR THE FIRST CALL.
C     INPUT-    ISWIT: A COUNTER, VERY FIRST CALL MUST HAVE ISWIT=0
```

```
C            X,Y,Z: GSM COORDINATES IN EARTH RADII
C            STAND: STAND-OFF DISTANCE IN RE
C            TILT: DIPOLE TILT ANGLE IN DEGREES
C
C     OUTPUT-   BXTOT,BYTOT,BZTOT: MAGNETIC FIELD COMPONENTS IN GAMMA
C               HJXS,HJYS,HJZS: CURRENT DENSITY COMPONENTS (AMPS/M**2)
C               THE INPUT VARIABLES FOR SUBROUT. BVOIGT,TAILC, AND
C                   RINGC ARE EXPLAINED IN EACH RESPECTIVE SUBROUT.
C
C     VARIABLES-XNST,ZNST: XGSM AND ZGSM POSITIONS OF THE CURRENT SHEET
C                          IN THE NOON-MIDNIGHT MERIDIAN. SEGMENT
C                          BOUNDAFORM XNST(I),ZNST(I)
C               BNST: CURRENT STRENGTH AT POINT XNST(I),ZNST(I) ABOVE
C               RADIUS: MAGNETOPAUSE RADIUS IN RE
C
C     SUBROUTINES-GETRAD: CALCULATES RADIUS FROM STAND-OFF DISTANCE
C                 BVOIOGT: FINDS VACUUM SHIELDED DIPOLE CONTRIBUTIONS
C                 ZPOS:  CALCULATES COEFFICIENTS FOR PLACEMENT OF TAIL
C                        CURRENT SHEET
C                 SEGMNT: FINDS XNST,ZNST, AND BNST FOR NSTEP SEGMENTS
C                         (NOT CALLED IF HJNEAR = 0)
C                 TAILC: FINDS TAIL CURRENT SHEET CONTRIBUTIONS
C                        (NOT CALLED IF HJNEAR = 0.)
C                 RINGC: FINDS RING CURRENT SYSTEM CONTRIBUTIONS
C                        (NOT CALLED IF BRN = BRP = 0.)
C                 TCOLLP: FINDS COLLAPSING TAIL CURRENT CONTRIBUTIONS
C                        (NOT CALLED IF COLLAP = 1.)
C
      DIMENSION XNST(50),ZNST(50),BNST(50)
C
C          FIRST CALL MUST HAVE ISWIT = 0
      ISWIT = ISWIT + 1
C      *** LIMIT MAX VALUE OF ISWIT TO 4 ***
      IF (ISWIT.GT.3) THEN
        ISWIT = 4
      END IF
C
C          GET BVOIGT COEFFICIENTS
      IF (ISWIT.EQ.1) THEN
C
C
C          SET CONSTANTS FOR VACUUM VOIGT SHIELDED DIPOLE MODEL
C          NO PENETRATION OF IMF
        ALFA   = 1.00
        CDIPOL = 0.0
        CIMF   = 0.0
        BIMF   = 0.0
        DIR    = 0.0
C
C             ZERO QUANTITIES
        BXV = 0.
        BYV = 0.
        BZV = 0.
        BXT = 0.
        BYT = 0.
        BZT = 0.
        BXTC = 0.
        BYTC = 0.
        BZTC = 0.
        BXR = 0.
        BYR = 0.
        BZR = 0.
C
C               GET RADIUS FROM STAND
```

```
         CALL GETRAD(STAND,RADIUS)
C                GET  COEFFICIENTS FOR BVOIGT
         CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
     1                  ISWIT,X,Y,Z,GARB1,GARB2,GARB3)
       END IF
C
C            SET CONSTANTS IN BVOIGT,SET-UP TAIL CURRENT WITH POSITION
       IF (ISWIT.EQ.2) THEN
           CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
     1                  ISWIT,X,Y,Z,GARB1,GARB2,GARB3)
C            CALCULATE COEFFICIENTS FOR CURRENT SHEET POSITION
         CALL ZPOS(ISWIT,TILT,X,Z)
C            ADVANCE ISWIT BECAUSE ZPOS CALLED NEXT IN SEGMNT
         ISWIT = ISWIT + 1
C            CALCULATE SEGMENT BOUNDARIES AND CURRENT STRENGTHS
         IF (HJNEAR.NE.0.) THEN
           CALL SEGMNT(ISWIT,TILT,XNEAR,HJNEAR,HJFRAC,NSTEP,
     1                  XNST,ZNST,BNST)
         END IF
       END IF
C
C            GET POINT VALUES
       IF (ISWIT.EQ.4) THEN
         CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
     1                  ISWIT,X,Y,Z,BXV,BYV,BZV)
         IF (HJNEAR.NE.0.) THEN
           CALL TAILC(ISWIT,NSTEP,X,Y,Z,RADIUS,DY,D,XNST,ZNST,BNST,
     1                 BXT,BYT,BZT)
           IF (COLLAP.NE.1.) THEN
             CALL TCOLLP(ISWIT,NSTEP,X,Y,Z,RADIUS,DYC,D,XNST,ZNST,
     1             BNST,COLLAP,DELXC,BXTC,BYTC,BZTC)
           END IF
         END IF
         IF ((BRN.NE.0.).OR.(BRP.NE.0.)) THEN
           CALL RINGC(X,Y,Z,TILT,BRN,BRP,RN,RP,BXR,BYR,BZR)
         END IF
C
           BXTOT = BXV + BXT + BXTC + BXR
           BYTOT = BYV + BYT + BYTC + BYR
           BZTOT = BZV + BZT + BZTC + BZR
C
       END IF
C
        RETURN
        END
C
C***************************** GETRAD
C
      SUBROUTINE GETRAD(STAND,RADIUS)
C
C     PURPOSE-  GIVES MAGNETOTAIL RADIUS AS A FUNCTION OF THE STAND-OFF
C               DISTANCE.
C     INPUT-    STAND: THE STAND-OFF DISTANCE (RE)
C     OUTPUT-   RADIUS: OF THE MAGNETOTAIL (RE)
C     VARIABLES-A0,A1,A2: COEFFICIENTS
C               MAXRAD: UPPER LIMIT RADIUS ALLOWED
C               MINRAD: LOWER LIMIT RADIUS ALLOWED
C               MXSTND: UPPER LIMIT STAND-OFF DISTANCE
C               MNSTND: LOWER LIMIT STAND-OFF DISTANCE
C     NOTE: DERIVED BY MATCHING STAND-OFF DISTANCES 7,11,14 RE WITH
C           RADII 17,20,27 RE RESPECTIVELY
C
      A0 = 29.1660
      A1 = -3.3214
```

```
      A2 =  0.2262
      MAXRAD = 27.
      MINRAD = 17.
      MXSTND = 14.
      MNSTND = 7.
C
C
      IF (STAND.GT.MXSTND) THEN
        RADIUS = MAXRAD
C       STAND  = MXSTND
      ELSE
        IF (STAND.LT.MNSTND) THEN
          RADIUS = MINRAD
C         STAND  = MNSTND
        ELSE
          RADIUS = A0 + A1*STAND + A2*(STAND**2)
        END IF
      END IF
C
      RETURN
      END
C*************************************  TAILC
      SUBROUTINE TAILC(ISWIT,NSTEP,XGSM,YGSM,ZGSM,RADIUS,DY,D,XNST,ZNST,
     1                 BNST,BX,BY,BZ)
C
C
C     PURPOSE-  CALCULATE THE MAGNETIC FIELD AND CURRENTS DUE TO A
C               CROSSTAIL CURRENT SHEET. MULTIPLE CURRENT SEGMENTS ARE
C               JOINED EDGE TO EDGE. THE CURRENT STRENGTH AND POSITION
C               AS A FUNCTION OF XGSM ARE ARBITRARY AND PROVIDED.
C               A YGSM DEPENDENCE IS ALSO INCLUDED.  INSPIRATION FOR
C               THIS MODEL CAN BE FOUND IN
C                    TSYGANENKO & USMANOV, PLANETARY AND SPACE SCIENCE,
C                      VOL. 30, PP. 985-998, 1982
C
C     INPUT-    ISWIT: COUNTER USED IN SUBROUT. ZPOS
C               NSTEP: NUMBER OF CURRENT SEGMENTS COMPRISING THE TAIL
C               XGSM,YGSM,ZGSM:  THE GEOCENTRIC SOLAR MAGNETOSPHERIC
C                                COORDINATE  (EARTH RADII)
C                     NOTE: ALL REFERENCES TO DISTANCE ARE MADE IN EARTH
C                     RADII IN THE GSM COORDINATE SYSTEM.
C               RADIUS: (RE) OF MAGNETOTAIL
C               DY: CHAR. WIDTH FOR Y DEPENDENCE OF BX AND BZ
C               D: HALF-WIDTH OF CURRENT FILAMENTS
C               XNST,ZNST: LOCATION OF CURRENT SHEET IN GSM X AND Z
C               BNST: CURRENT STRENGTH AT XNST,ZNST
C     OUTPUT-   BX,BY,BZ  MAGNETIC FIELD COMPONENTS IN GSM  (GAMMAS)
C               HJX,HJY,HJZ  CURRENT DENSITY IN GSM (AMPS/METER**2)
C
C     VARIABLES-XN,XF: NEAR AND FAR EDGES OF A CURRENT SEGMENT
C               ZN,ZF: NEAR AND FAR ZGSM POSITION OF CURRENT SEGMENT
C               BN,BF: VALUES OF B AT XN AND XF(REALLY CURRENT STRENGTH)
C               DELX: STEP SIZE = (XN-XF)
C               FY: VALUE OF Y-DEPENDENT FUNCTION (SEE SUBROUT. YYY)
C               FYP: FIRST DERIVATIVE OF Y-DEPENDENT FUNCTION ABOVE
C               DELB: =(BN-BF)
C               DD: = DELB/DELX
C               SLOPE: SLOPE OF A SEGMENT =(ZF-ZN)/(XF-XN)
C               RADIUS: RADIUS OF MAGNETOTAIL IN RE
C               FACTJ:  FACTOR TO GET CURRENT IN UNITS AMPS/METER**2
C                     = (1/MAGNETIC PERM)*(TESLA/GAMMA)*(RE/METER)
C                     = (1/(4*PI*1E-07))*(1E-09)*(1/(6.370E+06))
C                     = 1.249E-10
C               ALL OTHER VARIABLES ARE PRESENT FOR CONVENIENCE AND HAVE
```

```
C           NO PARTICULAR PHYSICAL MEANINGS IN MOST CASES.
C
C     SUBROUT.- YYY: RETURNS VALUE OF YGSM FUNCTIONAL DEPENDENCE AND THE
C                    FIRST DERIVATIVE W.R.T. YGSM AT A GIVEN YGSM
C
C
      DIMENSION XNST(50),ZNST(50),BNST(50)
      FACTJ = 1.249E-10
C
      BX = 0.
      BY = 0.
      BZ = 0.
C
C        CALCULATE CONTRIBUTIONS FROM NSTEP CURRENT SEGMENTS TO B-FIELD
      DO 10 N = 1, NSTEP
C
        XN = XNST(N)
        XF = XNST(N+1)
        DELX = XN - XF
        ZN = ZNST(N)
        ZF = ZNST(N+1)
        SLOPE = (ZF - ZN)/(XF - XN)
C
        BN = BNST(N)
        BF = BNST(N+1)
        DELB = BN - BF
C
        DD = DELB/DELX
        QB = BN - DD*XN
C
C           CONSTANTS INVOLVED IN INTEGRATION: A,B,C ARE COMMON TO BOTH
C                      THE BX AND BZ EXPRESSIONS, BY = 0
C
        A = 1. + SLOPE**2
        B = -2.*(XGSM + SLOPE*(ZGSM - ZN + SLOPE*XN))
        C = XGSM**2 + (ZGSM - ZN + SLOPE*XN)**2 + D**2
        BDX = -2.
        BDZ = -2.*SLOPE
        CDX = 2.*XGSM
        CDZ = 2.*(ZGSM - ZN + SLOPE*XN)
C
C         TEST FOR PROPER INTEGRAL FORM, ONLY THE FORM FOR ABC > 0 HAVE
C             BEEN INCLUDED. NO EXCEPTIONS FOUND TO DATE
C
        ABC = 4.*A*C - B**2
        SABC = SQRT(ABC)
        IF (ABC.LE.0.) THEN
          GOTO 15
        END IF
C
C           SOME  CONSTANTS FOR BX EXPRESSION
C
        CMX = -SLOPE*DD
        CNX = -SLOPE*QB + DD*(SLOPE*XN + ZGSM - ZN)
        CPX = (SLOPE*XN + ZGSM - ZN)*QB
C
C           SOME  CONSTANTS FOR BZ EXPRESSION
C
        CMZ = DD
        CNZ = BN - DD*(XN + XGSM)
        CPZ = -XGSM*QB
C
C           SOME  COMMON FACTORS FOR BOTH BX AND BZ
C
```

```
         WLN = LOG((A*XN**2+B*XN+C)/(A*XF**2+B*XF+C))
         WATAN = ATAN((2*A*XN+B)/SABC) - ATAN((2*A*XF+B)/SABC)
C
C           THE  BX FUNCTION
C
         CLNX = (A*CNX - B*CMX)/(2.*(A**2))
         CTANX = (2.*A*(A*CPX-C*CMX)-B*(A*CNX-B*CMX))/((A**2)*SABC)
         BX = BX + (CMX*DELX)/A + CLNX*WLN + CTANX*WATAN
C
C           THE BZ FUNCTION
C
         CLNZ = (A*CNZ - B*CMZ)/(2.*(A**2))
         CTANZ = (2.*A*(A*CPZ-C*CMZ)-B*(A*CNZ-B*CMZ))/((A**2)*SABC)
         BZ = BZ + (CMZ*DELX)/A + CLNZ*WLN + CTANZ*WATAN
C
 10   CONTINUE
C
C          GET YGSM DEPENDENCE AND FIRST DERIVATIVE
      CALL YYY(YGSM,DY,RADIUS,FY,FYP)
C
C          NOW INCLUDE THE YGSM DEPENDENCE IN B-FIELD AND CURRENT
      BX = BX*FY
      BY = 0.
      BZ = BZ*FY
      GOTO 20
C
 15   CONTINUE
      WRITE(6,99)
 99   FORMAT(1H1,'WRONG EQUATION USED, OUTPUT SET TO ZERO')
      BX = 0.
      BY = 0.
      BZ = 0.
C
 20   CONTINUE
      RETURN
      END
C
C******************************** HJSIZE
C
       FUNCTION HJSIZE(X,XNEAR,HJNEAR,HJFRAC)
C
C     PURPOSE-  CALCULATE RELATIVE CURRENT STRENGTH ALONG MIDNIGHT
C               MERIDIAN AT A GIVEN XGSM POSITION. FUNCTION IS ARBITRARY
C     INPUT-    X: XGSM POSITION
C               XNEAR: TAIL CURRENT INNER EDGE DISTANCE (NEG. TAILWARD)
C               HJNEAR: HJSIZE AT XNEAR
C               HJFRAC: FRACTION OF HJNEAR FOUND AT -100 RE
C     OUTPUT-   HJSIZE: STRENGTH OF CROSSTAIL CURRENT
C     VARIABLES-BFLAT: X POSITION WHERE HJSIZE LEVELS OFF TO CONSTANT RE
C                 NOTE: BFLAT IS SET AT -100. RE, AN AD HOC VALUE
C               EXPON: EXPONENTIAL DECREASE OF CURRENT DENSITY
C
       BFLAT = -100.
       EXPON = LOG(HJFRAC)/(BFLAT - XNEAR)
       IF (X.GT.XNEAR) THEN
           HJSIZE = 0.
           GOTO 10
       END IF
C
       IF (X.GT.(BFLAT)) THEN
           HJSIZE = HJNEAR*EXP(EXPON*(X - XNEAR))
       ELSE
           HJSIZE = HJNEAR*HJFRAC
       END IF
```

```
C
 10     CONTINUE
        RETURN
        END
C
C**************************************  ZPOS
C
      SUBROUTINE ZPOS(ISWIT,TILT,XGSM,ZGSM)
C
C         PURPOSE-  CALCULATE ZGSM POSITION OF CURRENT SHEET IN
C                        MIDNIGHT MERIDIAN
C         INPUT-    TILT: DIPOLE TILT ANGLE
C                   ISWIT: COUNTER SO COEFFICIENTS ONLY CALCULATED ONCE
C                   XGSM: COORDINATE OF INTEREST
C         OUTPUT-   ZGSM: CURRENT SHEET ELEVATION W.R.T. THE XGSM AXIS
C         VARIABLES-C1: XGSM WHERE CURVE LEVELS OFF, ZERO SLOPE
C                   C2: XGSM WHERE CURVE LEAVES XSM AXIS
C                   C3: ZGSM AT XGSM=C2
C                   C4: ZGSM AT XGSM>=C1
C                   C5: SLOPE OF XSM AXIS IN GSM FRAME AT C2
C                   RTILT: DIPOLE TILT IN RADIANS
C                   HINGE: CURRENT SHEET HINGE POSITION, SET = 9. RE
C                             AFTER GOSLING, JGR, 7093, 1986
C                   A0,A1,A2,A3: COEFFICIENTS OF POLYNOMIAL FIT
C                        Z = A0 + A1*X + A2*X**2 + A3*X**3
C
      IF (ISWIT.LE.2) THEN
C         CALCULATE COEFFICIENTS
        HINGE = 9.
        DEG = ACOS(-1.00)/180.
        RTILT = DEG*TILT
C
        C1 = 4. - HINGE*(1. + COS(RTILT))
        C2 = -4.*COS(RTILT)
        C3 =  4.*SIN(RTILT)
        C4 =  HINGE*SIN(RTILT)
        C5 = -TAN(RTILT)
C
        Q1 = 3.*(C1**2)*C2 - C2**3
        Q2 = -2.*C1*C2 + C2**2
        Q3 = C2 - C1
        Q4 = C2**2 - C1**2
C
        A3 = (2.*(C4-C3)+C5*Q3)/(3.*Q3*Q4+2.*(Q1-2.*(C1**3)))
        A2 = (C5 - 3.*A3*Q4)/(2.*Q3)
        A1 = -2.*A2*C1 - 3.*A3*(C1**2)
        A0 = C3 - A2*Q2 + A3*Q1
      END IF
C
C         INSIDE CURVE LIMIT USE XSM AXIS
      IF (XGSM.GT.C2) THEN
        ZGSM =  XGSM*C5
      END IF
C
C         BETWEEN XGSM C2 AND C1, FITTED CURVE IS USED
      IF ((XGSM.LE.C2).AND.(XGSM.GE.C1)) THEN
        ZGSM = A0 + A1*XGSM + A2*(XGSM**2) + A3*(XGSM**3)
      END IF
C
C         OUTSIDE CURVE LIMIT USE LINE PARALLEL TO XGSM AXIS
      IF (XGSM.LT.C1) THEN
        ZGSM = C4
      END IF
C
```

```
      RETURN
      END
C
C********************************** YYY
      SUBROUTINE YYY(Y,DY,RAD,FY,FYP)
C
C     PURPOSE-  CALCULATE YGSM DEPENDENCE AND FIRST DERIVATIVE
C               FY = 0 AT Y = RADIUS
C     INPUT-    Y: YGSM  POSITION (EARTH RADII)
C               RAD: RADIUS OF MAGNETOTAIL
C               DY: CHARACTERISTIC WIDTH OF Y DEPENDENCE
C     OUTPUT-   FY: FUNCTIONAL VALUE OF YGSM DEPENDENCE AT Y
C               FYP: FIRST DERIVATIVE OF FY W.R.T Y
C
      YEXP = -(Y/DY)**2
      IF (YEXP.LT.(-40.)) THEN
        FY = 0.
        FYP = 0.
        GOTO 10
      END IF
C
      PI = ACOS(-1.00)
      CY = COS(PI*Y/(2.*RAD))
      SY = SIN(PI*Y/(2.*RAD))
C
      FY  = CY*EXP(-(Y/DY)**2)
      FYP = (-2.*Y*CY/(DY**2) - PI*SY/(2.*RAD))*EXP(-(Y/DY)**2)
 10   CONTINUE
C
      RETURN
      END
C
C********************************** RINGC
C
      SUBROUTINE RINGC(XGSM,YGSM,ZGSM,TILT,BRN,BRP,RN,RP,
     1                 BXGSM,BYGSM,BZGSM)
C
C     PURPOSE-  CALCULATE MAGNETIC FIELD AND CURRENT COMPONENTS OF RING
C               CURRENT TILTED WITH THE DIPOLE
C     INPUT-    XGSM,YGSM,ZGSM: GSM COORDINATES IN RE
C               TILT: TILT IN DEGREES OF CENTRAL AXIS OF RINGS, ALIGNED
C                     WITH DIPOLE AXIS. POSITIVE FOR N. HEMISPHERE SUMM.
C               BRN: NEGATIVE BZ CONTRIBUTION FOR ZERO TILT AT (0,0,0)
C                    FROM THE WESTWARD CURRENT
C               BRP: POSITIVE BZ CONTRIBUTION FROM EASTWARD CURRENT
C               RN:  CHARACTERISTIC RADIUS (IN RE) OF WESTWARD RING
C               RP:  CHARACTERISTIC RADIUS (IN RE) OF EASTWARD RING
C               BXGSM,BYGSM,BZGSM: MAGNETIC FIELD COMPONENTS OF RING
C                                  IN GAMMAS
C     OUTPUT-   FACTJ: CONSTANT USED TO ADJUST UNITS OF CURRENT, SEE
C     VARIABLES-BR:    SUBROUT. TAILC FOR DERIVATION
C               RHO,PHI,Z: CYLINDRICAL COORDINATE SYSTEM
C     SUBROUTINES- TRANS6: CHANGES GSM CARTESIAN TO SM CYLINDRICAL
C                  VTRAN6: CHANGES SM CYLINDRICAL VECTORS TO GSM SYSTEM
C
C
C          FACTOR TO MAKE CURRENT OUTPUT IN AMPS/M**2
      FACTJ = 1.249E-10
      PI = ACOS(-1.00)
C
C          CHANGE GSM TO SM CYLINDRICAL COORDINATES
      CALL TRANS6(TILT,XGSM,YGSM,ZGSM,RHO,PHI,Z)
C
C          SCALE FACTOR (PUT Z AND RHO INTO UNITS OF CHAR. RADII)
```

```
      ZN = Z/RN
      RHON = RHO/RN
      ZP = Z/RP
      RHOP = RHO/RP
C
C          COMMON FACTORS
      AN = RHON**2 + ZN**2 + 4.
      BN = 2.*(ZN**2) - (RHON**2) + 8.
      AP = RHOP**2 + ZP**2 + 4.
      BP = 2.*(ZP**2) - (RHOP**2) + 8.
C
C          CYLINDRICAL BFIELD
      BRHO = 12.*BRN*RHON*ZN/SQRT(AN**5) + 12.*BRP*RHOP*ZP/SQRT(AP**5)
      BPHI = 0.
      BZ   = 4.*BRN*BN/SQRT(AN**5) + 4.*BRP*BP/SQRT(AP**5)
C
C          CHANGE VECTOR QUANTITIES BACK TO GSM COORDINATES
      CALL VTRAN6(BRHO,BPHI,BZ,PHI,TILT,BXGSM,BYGSM,BZGSM)
C
      RETURN
      END
C
C
C*************************** TRANS6
      SUBROUTINE TRANS6(TILT,XGSM,YGSM,ZGSM,RHO,PHI,Z)
C
C          PURPOSE-  CHANGE GSM CARTESIAN TO SM CYLINDRICAL COORDINATES
C          INPUT-    TILT: DIPOLE TILT IN DEGREES
C                    XGSM,YGSM,ZGSM: GSM COORDINATES
C          OUTPUT-   RHO,PHI,Z: CYLINDRICAL COORDINATES OF SM SYSTEM
C                    UNITS (RE,RADIANS,RE)
C          VARIABLES-RTILT: TILT ANGLE IN RADIANS
C
      PI = ACOS(-1.00)
      RTILT = TILT*(PI/180.)
C
C          CHANGE GSM TO SM COORDINATES
      XSM = XGSM*COS(RTILT) - ZGSM*SIN(RTILT)
      YSM = YGSM
      ZSM = XGSM*SIN(RTILT) + ZGSM*COS(RTILT)
C
      RHO = SQRT(XSM**2 + YSM**2)
      IF (RHO.EQ.0.) THEN
        RHO = 0.000001
      END IF
      PHI = ACOS(XSM/RHO)
      IF (YSM.LT.0.) THEN
        PHI = PHI + 2.*(PI - PHI)
      END IF
      Z   = ZSM
C
      RETURN
      END
C
C*********************  VTRAN6
C
      SUBROUTINE VTRAN6(VRHO,VPHI,VZ,PHI,TILT,VXGSM,VYGSM,VZGSM)
C
C          PURPOSE-  CHANGE VECTOR QUANTITIES OF SM CYLINDRICAL TO
C                    VECTORS OF GSM COORDINATES
C          INPUT-    VRHO,VPHI,VZ: RHO,PHI,Z VECTORS
C                    PHI: ANGLE PHI IN RADIANS
C                    TILT: DIPOLE TILT ANGLE IN DEGREES
C          OUTPUT-   VXGSM,VYGSM,VZGSM: GSM VECTOR COMPONENTS
```

```
C         VARIABLES-RTILT: DIPOLE TILT ANGLE IN RADIANS
C
      PI = ACOS(-1.00)
      RTILT = TILT*(PI/180.)
C
C         CHANGE CYLINDRICAL TO CARTESIAN (BOTH SM)
      VXSM = VRHO*COS(PHI) - VPHI*SIN(PHI)
      VYSM = VRHO*SIN(PHI) + VPHI*COS(PHI)
      VZSM = VZ
C
C         CHANGE SM TO GSM
      VXGSM =  VXSM*COS(RTILT) + VZSM*SIN(RTILT)
      VYGSM =  VYSM
      VZGSM = -VXSM*SIN(RTILT) + VZSM*COS(RTILT)
C
      RETURN
      END
C

C**************************************** TCOLLP
      SUBROUTINE TCOLLP(ISWIT,NSTEP,XGSM,YGSM,ZGSM,RADIUS,DYC,D,
     1                 XNST,ZNST,BNST,COLLAP,DELXC,BX,BY,BZ)
C
C
C    ***** SPECIAL COLLAPSE OF TAIL SIMUALTION, INTRODUCES LESSER
C          CURRENT IN OPPOSITE DIRECTION AT NEAR PART OF TAIL, ADDED TO
C          REGULAR CURRENT.
C    *****
C     PURPOSE-  CALCULATE THE MAGNETIC FIELD AND CURRENTS DUE TO A
C               CROSSTAIL CURRENT SHEET. MULTIPLE CURRENT SEGMENTS ARE
C               JOINED EDGE TO EDGE. THE CURRENT STRENGTH AND POSITION
C               AS A FUNCTION OF XGSM ARE ARBITRARY AND PROVIDED.
C               A YGSM DEPENDENCE IS ALSO INCLUDED.
C
C     INPUT-    ISWIT: COUNTER USED IN SUBROUT. ZPOS
C               NSTEP: NUMBER OF CURRENT SEGMENTS IN TAIL
C               XGSM,YGSM,ZGSM:  THE GEOCENTRIC SOLAR MAGNETOSPHERIC
C                                COORDINATE  (EARTH RADII)
C                   NOTE: ALL REFERENCES TO DISTANCE ARE MADE IN EARTH
C                   RADII IN THE GSM COORDINATE SYSTEM.
C               DYC: CHAR. WIDTH FOR Y DEPENDENCE OF BX AND BZ
C               D: HALF-WIDTH OF CURRENT FILAMENTS
C               XNST,ZNST: XGSM AND ZGSM POSITION OF CURRENT SHEET
C               BNST: CURRENT STRENGTH AT ANST,ZNST
C               COLLAP: FRACTION OFT AIL FIELD CURRENT LIET AT MIDNIGHT
C                    = 1. MEANS NO CHANGE, ORIGINAL TAIL ONLY
C                    = 0. MEANS COMPLETE COLLAPSE AT INNER EDGE
C               DELXC: LENGTH OF INNER TAIL CURRENT SHEET (IN X) THAT IS
C                    DISRUPTED
C
C     OUTPUT-   BX,BY,BZ  MAGNETIC FIELD COMPONENTS IN GSM  (GAMMAS)
C
C     VARIABLES-XN,XF: NEAR AND FAR EDGES OF A CURRENT SEGMENT
C               BN,BF: VALUES OF B AT XN AND XF(REALLY CURRENT STRENGTH)
C               NSTEP: NUMBER OF CURRENT SEGMENTS IN TAIL WITHIN 100 RE
C               DELX: STEP SIZE = (XN-XF)
C               FY: VALUE OF Y-DEPENDENT FUNCTION (SEE SUBROUT. YYY  )
C               FYP: FIRST DERIVATIVE OF Y-DEPENDENT FUNCTION ABOVE
C               DELB: =(BN-BF)
C               DD: = DELB/DELX
C               SLOPE: SLOPE OF A SEGMENT =(ZF-ZN)/(XF-XN)
C                    = 1.249E-10
C               ALL OTHER VARIABLES ARE PRESENT FOR CONVENIENCE AND HAVE
C               NO PARTICULAR PHYSICAL MEANINGS IN MOST CASES.
```

```
C
C     SUBROUT.- YYY: RETURNS YGSM FUNCTIONAL DEPENDENCE AND THE
C                    FIRST DERIVATIVE W.R.T. YGSM AT A GIVEN YGSM
C
      DIMENSION XNST(50),ZNST(50),BNST(50)
C
      BX = 0.
      BY = 0.
      BZ = 0.
C
C        CALCULATE CONTRIBUTIONS FROM NSTEP CURRENT SEGMENTS TO B-FIELD
C
      DO 10 N = 1, NSTEP
C
        XN = XNST(N)
        XF = XNST(N+1)
        DELX = XN - XF
C
        ZN = ZNST(N)
        ZF = ZNST(N+1)
        SLOPE = (ZF - ZN)/(XF - XN)
C
        BN = BNST(N)
        BF = BNST(N+1)
C         ADJUST CURRENT SIZE AND DIRECTION FROM ORIGINAL
C                (0.1) FACTOR BELOW GIVES 5 RE COLLAPSE "EDGE"
        CFN = .5*(TANH(0.1*(-XN+XNST(1)-DELXC))-TANH(-0.1*DELXC) - 2.)
        CFF = .5*(TANH(0.1*(-XF+XNST(1)-DELXC))-TANH(-0.1*DELXC) - 2.)
C         NO NEED TO CALCULATE COLLAPSE CURRENT FOR FAR TAIL
C         CUT OUT OF LOOP WHEN CONTRIBUTION IS LESS THAN 2 PERCENT
C         THE MAGNITUDE OF THE REGULAR CURRENT OF TAILC
        IF (CFF.GT.(-0.02)) THEN
          GOTO 11
        END IF
        BN = (1.- COLLAP)*BN*CFN
        BF = (1.- COLLAP)*BF*CFF
C
        DELB = BN - BF
        DD = DELB/DELX
        QB = BN - DD*XN
C
C            CONSTANTS INVOLVED IN INTEGRATION: A,B,C ARE COMMON TO BOTH
C                       THE BX AND BZ EXPRESSIONS, BY = 0
        A = 1. + SLOPE**2
        B = -2.*(XGSM + SLOPE*(ZGSM - ZN + SLOPE*XN))
        C = XGSM**2 + (ZGSM - ZN + SLOPE*XN)**2 + D**2
        BDX = -2.
        BDZ = -2.*SLOPE
        CDX = 2.*XGSM
        CDZ = 2.*(ZGSM - ZN + SLOPE*XN)
C
C          TEST FOR PROPER INTEGRAL FORM, ONLY THE FORM FOR ABC > 0 HAVE
C                BEEN INCLUDED. NO EXCEPTIONS FOUND TO DATE
        ABC = 4.*A*C - B**2
        SABC = SQRT(ABC)
        IF (ABC.LE.0.) THEN
          GOTO 15
        END IF
C
C            SOME  CONSTANTS FOR BX EXPRESSION
        CMX = -SLOPE*DD
        CNX = -SLOPE*QB + DD*(SLOPE*XN + ZGSM - ZN)
        CPX = (SLOPE*XN + ZGSM - ZN)*QB
C
```

```
C             SOME  CONSTANTS FOR BZ EXPRESSION
          CMZ = DD
          CNZ = BN - DD*(XN + XGSM)
          CPZ = -XGSM*QB
C
C             SOME  COMMON FACTORS FOR BOTH BX AND BZ
C
          WLN = LOG((A*XN**2+B*XN+C)/(A*XF**2+B*XF+C))
          WATAN = ATAN((2*A*XN+B)/SABC) - ATAN((2*A*XF+B)/SABC)
C
C             THE  BX FUNCTION
          CLNX = (A*CNX - B*CMX)/(2.*(A**2))
          CTANX = (2.*A*(A*CPX-C*CMX)-B*(A*CNX-B*CMX))/((A**2)*SABC)
          BX = BX + (CMX*DELX)/A + CLNX*WLN + CTANX*WATAN
C
C             THE BZ FUNCTION
          CLNZ = (A*CNZ - B*CMZ)/(2.*(A**2))
          CTANZ = (2.*A*(A*CPZ-C*CMZ)-B*(A*CNZ-B*CMZ))/((A**2)*SABC)
          BZ = BZ + (CMZ*DELX)/A + CLNZ*WLN + CTANZ*WATAN
C
 10    CONTINUE
 11    CONTINUE
C
C           USING SAME Y FUNCTION AS TAILC SUBROUT., BUT WITH DIFFERENT
C           DY VALUE (DYC HERE)
C           GET YGSM DEPENDENCE AND FIRST DERIVATIVE
       CALL YYY(YGSM,DYC,RADIUS,FY,FYP)
C
C           NOW INCLUDE THE YGSM DEPENDENCE IN B-FIELD AND CURRENT
        BX = BX*FY
        BY = 0.
        BZ = BZ*FY
        GOTO 20
C
 15     CONTINUE
        WRITE(6,99)
 99     FORMAT(1H1,'WRONG EQUATION USED, OUTPUT SET TO ZERO')
        BX = 0.
        BY = 0.
        BZ = 0.
C
 20     CONTINUE
        RETURN
        END
C********************** SEGMNT
       SUBROUTINE SEGMNT(ISWIT,TILT,XNEAR,HJNEAR,HJFRAC,NSTEP,XNST,
     1                      ZNST,BNST)
C
C      PURPOSE-  FIND THE GSM X AND Z LOCATIONS OF THE SEGMENT BOUNDARIES
C                USED TO BUILD THE TAIL CURRENT. ALSO ASSIGN A CURRENT
C                STRENGTH FOR EACH POINT.  THE INNER EDGE IS ABS(XNEAR)
C                AWAY FROM EARTH.
C
C      INPUT-    ISWIT: COUNTER USED IN SUBROUT. ZPOS
C                TILT: DIPOLE TILT IN DEGREES
C                XNEAR: NEGATIVE DISTANCE TAILWARD,LOCATION OF INNER EDGE
C                HJNEAR: CURRENT STRENGTH AT INNER EDGE
C                HJFRAC: FRACTION OF HJNEAR REMAINING IN FAR TAIL
C      OUTPUT-   NSTEP: NUMBER OF SEGMENTS USED TO BUILD THE TAIL CURRENT
C                XNST,ZNST: GSM X AND Z LOCATIONS SEGMENT BOUNDARIES
C                BNST:  CURRENT STRENGTH AT (XNST,ZNST)
C      VARIABLES-FRAC: FRACTION OF DESIRED SEGMENT LENGTH
C                PI: JUST PI
C                RTILT: TILT ANGLE IN RADIANS
```

```
C              XNGSM,ZNGSM: TEST XGSM AND ZGSM TO FIND INNER EDGE
C              SN: DISTANCE ALONG SHEET PLUS INNER EDGE DISTANCE
C              DELS: DESIRED SEGMENT LENGTH
C              DIST: TEST DISTANCE COMPARED TO DELS OR ABS(XNEAR)
C
C     SUBROUT.- ZPOS: FINDS ZGSM LOCATION OF CURRENT SHEET AT ANY XGSM
C              HJSIZE: FINDS CURRENT STRENGTH AT POINT ALONG SHEET
C
      DIMENSION XNST(50),ZNST(50),BNST(50)
C
      NSTEP = 16
C           TAIL IS 350 RE LONG, MADE OF 16 SEGMENTS OF VARYING LENGTHS
      FRAC = 0.025
C          SET INNER EDGE POINT, ABS(XNEAR) IS DISTANCE FROM EARTH
      IF (TILT.NE.0.) THEN
C          STEP ALONG XGSM AXIS UNTIL PROPER DISTANCE IS FOUND
        PI = ACOS(-1.00)
        RTILT = TILT*PI/180
        XNGSM = 0.0
 2      CONTINUE
        XNGSM = XNGSM - 0.05
        CALL ZPOS(ISWIT,TILT,XNGSM,ZNGSM)
        DIST = SQRT(XNGSM**2 + ZNGSM**2)
C          SEE IF INNER EDGE IS AT PROPER DISTANCE FROM EARTH
        IF (DIST.GE.ABS(XNEAR)) THEN
          XNST(1) = XNGSM
          ZNST(1) = ZNGSM
        ELSE
          GOTO 2
        END IF
      ELSE
C          TILT = 0
        XNST(1) = XNEAR
        ZNST(1) = 0.
      END IF
      SN = XNST(1)
      BNST(1) = HJSIZE(SN,XNST(1),HJNEAR,HJFRAC)
C          GET OTHER SEGMENT BOUNDARY POINTS
      DO 10 N = 1,NSTEP
        I = N + 1
C          GET STEP SIZE
        IF (N.LE.8) THEN
          DELS = 2.0
        ELSE
          IF (N.LE.11) THEN
            DELS = 5.
          ELSE
            IF (N.LE.14) THEN
              DELS = 10.
            ELSE
              IF (N.EQ.15) THEN
                DELS = 39.
              END IF
            END IF
          END IF
        END IF
        IF (N.EQ.NSTEP) THEN
          DELS = 250.
        END IF
C
        SN = SN - DELS
        BNST(I) = HJSIZE(SN,XNST(1),HJNEAR,HJFRAC)
        IF (TILT.NE.0.) THEN
          IF (I.GT.2) THEN
```

```
          IF (ZNST(I-1).EQ.ZNST(I-2)) THEN
C               NOW IN FLAT PART OF CURRENT SHEET
            XNST(I) = XNST(I-1) - DELS
            ZNST(I) = ZNST(I-1)
            GOTO 9
          END IF
        END IF
        XNST(I) = XNST(I-1)
 5      CONTINUE
        XNST(I) = XNST(I) - FRAC*DELS
        CALL ZPOS(ISWIT,TILT,XNST(I),ZNST(I))
        DIST = SQRT((XNST(I)-XNST(I-1))**2+(ZNST(I)-ZNST(I-1))**2)
        IF (DIST.GE.DELS) THEN
C               THEN NEXT SEGMENT BOUNDARY IS FOUND
          GOTO 9
        ELSE
C               ADD ANOTHER PIECE
          GOTO 5
        END IF
      ELSE
C          TILT IS ZERO
        XNST(I) = SN
        ZNST(I) = 0.
      END IF
 9    CONTINUE
 10   CONTINUE
      RETURN
      END
C ***********************************:      BVOIGT
      SUBROUTINE BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF BIMF,DIR,
     1                  ISWIT, X,Y,Z, BX,BY,BZ)
C
C   PURPOSE-     COMPUTES THREE COMPONENTS OF THE VECTOR MAGNETIC FIELD
C                AT ANY POINT. POINTS OUTSIDE OF THE MAGNETOPAUSE WILL
C                ONLY RETURN NONZERO FIELDS IF INTERCONNECTION IS ALLOWED
C                WITH THE IMF.
C
C   INPUT-       STAND,  THE SUBSOLAR STAND-OFF DISTANCE (IN EARTH RADII)
C                RADIUS, RADIUS OF THE MAGNETOTAIL (IN EARTH RADII)
C                TILT,   DIPOLE TILT ANGLE  (IN DEGREES)
C                        TILT>0 INDICATES NORTHERN HEMISPHERE SUMMER
C                ALFA,   PARAMETER INDICATING PLASMA CONTENT OF TAIL,
C                        = 1.0 MEANS VACUUM, = 0.0 MEANS HARRIS SHEET
C                CDIPOL, FRACTION OF DIPOLE FIELD ALLOWED TO PENETRATE
C                        OUT THROUGH THE MAGNETOPAUSE
C                CIMF,   FRACTION OF THE INTERPLANETARY FIELD ALLOWED TO
C                        PENETRATE IN THROUGH THE MAGNETOPAUSE
C                BIMF,   MAGNITUDE OF THE INTERPLANETARY MAGNETIC FIELD
C                        (IN GAMMA)
C                DIR,    DIRECTION OF IMF, 0 DEGREES IS NORTH, 90 DEGREES
C                        IS ANTISUNWARD
C                ISWIT,  INDICATES THE PROCEDURE TO FOLLOW ON THIS CALL
C                        BFIELD IS CALLED THREE TIMES WITH ISWIT = 1,2
C                        THEN EITHER 3 OR 4
C                 *************ALTERED SO >2 GIVES GSM RESULT 10/11/87
C                        = 1 GENERAL COEFFICIENTS ARE FOUND TO REPRESENT
C                          THE MAGNETIC FIELD WITH LEGENDRE AND BESSEL
C                          SERIES
C                        = 2 CONSTANTS SPECIFY THE CONFIGURATION DESIRED
C                          ARE CALCULATED AND STORED, I.E. TILT
C                        = 3 CALCULATES FIELD COMPONENTS, INPUT AND OUTPU
C                          ARE IN THE SM (SOLAR MAGNETOSPHERIC) SYSTEM
C                        = 4 SAME AS 3 BUT GSM COORDINATES ARE USED
C                 X,Y,Z  CARTESIAN COORDINATE IN EITHER SM OR GSM SYSTEM
```

```
C
C   OUTPUT-   BX,BY,BZ  MAGNETIC FIELD VECTOR IN CARTESIAN SYSTEM
C                       ACCORDING TO CHOICE ISWIT=3 OR 4. (IN GAMMA)
C
C
C   VARIABLES- NDIM   NUMBER OF GRID POINTS TO BE USED FOR INTEGRATION
C                     TO OBTAIN SERIES COEFFICIENTS
C              NELE   NUMBER OF ITERATIONS TO BE MADE BETWEEN SPHERICAL
C                     AND CYLINDRICAL COEFFICIENT CALCULATIONS.
C               BEX   DISTANCE BETWEEN SPHERE CENTER AND EARTH
C          XD,YD,ZD   CARTESIAN COORDINATES IN DIPOLE SYSTEM, XD IS
C                     PERPEND. TO DIPOLE MOMENT POINTING ANTISUNWARD
C       HXD,HYD,HZD   MAGNETIC FIELD COMPONENTS IN DIPOLE SYSTEM
C          XM,YM,ZM   EARTH CENTERED CARTESIAN SYSTEM, XM POINTS ANTI-
C                     SUNWARD, PARALLEL TO THE TAIL AXIS (CALLED THE M
C                     SYSTEM)
C       HXM,HYM,HZM   MAGNETIC FIELD COMPONENTS ON M SYSTEM
C         SNPS,CSPS   SIN AND COS OF THE DIPOLE TILT ANGLE
C
C    REFERENCES-
C
C          REF1:           VOIGT,G.-H., A MATHEMATICAL MAGNETOSPHERIC
C        FIELD MODEL WITH INDEPENDENT PHYSICAL PARAMETERS, PLANETARY
C        AND SPACE SCIENCE, VOL. 29, 1-20,1981
C
C          REF2:           ABRAMOWITZ AND STEGUN,EDS., HANDBOOK OF
C        MATHEMATICAL FUNCTIONS, U.S. NATIONAL BUREAU OF STANDARDS,
C        DOVER, NEW YORK, (1965)
C
C          REF3:           JAHNKE-EMDE-LOSCH, TABLES OF HIGHER
C        FUNCTIONS, 6TH EDITION(REVISED), NEW YORK, (1960)
C
C          REF4:           VOIGT PH.D. THESIS
C
C    COORDINATE SYSTEMS-
C
C        GSM       GEOCENTRIC SOLAR MAGNETOSPHERIC - EARTH CENTERED
C
C                  XGSM : FROM EARTH TO SUN
C                  YGSM : PERPENDICULAR TO EARTH'S DIPOLE, POINTING
C                         TOWARDS DUSK
C                  ZGSM : SAME SENSE AS NORTHERN MAGNETIC POLE
C                         DIPOLE CONTAINED WITHIN X-Z PLANE
C
C        SM        SOLAR MAGNETIC - EARTH CENTERED
C
C                  XSM : TOWARD SUN BUT NOT DIRECTLY UNLESS TILT ZERO
C                  YSM = YGSM
C                  ZSM : PARALLEL TO NORTH MAGNETIC POLE
C
C        M         THE M SYSTEM  (XM,YM,ZM) - EARTH CENTERED
C
C                  XM = -XGSM
C                  YM = -YGSM
C                  ZM =  ZGSM
C
C        V         THE V SYSTEM  (XV,YV,ZV) - SPHERE CENTERED
C
C                  XV = ZGSM
C                  YV = YGSM
C                  ZV = -XGSM - B  WHERE B = (TAIL RADIUS)-(STAND-OFF)
C
C        D         THE D SYSTEM  (XD,YD,ZD) - EARTH CENTERED
C
```

```
C                   XD = -XSM
C                   YD = -YSM = -YGSM
C                   ZD = ZSM
C
C         SPHERICAL SYSTEM   (R,THETA,PHI) - SPHERE CENTERED
C
C                   R : DISTANCE FROM ORIGIN (XV,YV,ZV)=(O,O,O)
C                   THETA : ANGLE BETWEEN ZV AND VECTOR R
C                   PHI : ANGLE BETWEEN XV AND COMPONENT OF VECTOR R
C                         IN THE XV-YV PLANE
C
C         CYLINDRICAL SYSTEM  (RHO,PHI,Z) - SPHERE CENTERED
C
C                   RHO : RADIAL DISTANCE FROM ZV AXIS
C                   PHI : ANGLE BETWEEN XV-ZV PLANE AND VECTOR RHO
C                             (SAME AS PHI OF SPHERICAL)
C                   Z = ZV
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      COMMON /BLOCK3/ H0, ALFB, N
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
      COMMON /FIMF/   HIMF, CD, CI, XI
C
      ALFB = ALFA
C
      R  = RADIUS
      CD = CDIPOL
      CI = CIMF
      HIMF = BIMF
      XI   = DIR
C
C
      IF (ISWIT.EQ.1) THEN
C              GET COEFFICIENTS FOR SPHERICAL AND BESSEL HARMONICS
          BEX = RADIUS - STAND
          NDIM = 101
          NELE = 4
          CALL COEFFS(NDIM,NELE)
      END IF
C
      IF (ISWIT.EQ.2) THEN
C              SET CONSTANTS, SPECIFY TILT WHICH WAS PRESET IN COEFFS
C              NOTE THAT SIGN OF TILT ANGLE IS BEING SWITCH
          TI = - TILT
          CALL CONST(TI)
      END IF
C
C          SM COORDINATE OPTION CUTOFF FOR NOW EVERYTHING IN GSM
CCC   IF (ISWIT.EQ.3) THEN
C              GET BFIELD AT POINT USING SM COORDINATE SYSTEM
C                   TRANSFORM SM TO DIPOLE SYSTEM, XD IS PERPENDICULAR
C                   TO DIPOLE MOMENT POINTING OPPOSITE XSM
CCC       XD = - X
CCC       YD = - Y
CCC       ZD = + Z
C
C                  TRANSFORM DIPOLE TO M SYSTEM, XM ANTISOLAR
CCC       CALL TRANS4(XD,YD,ZD,XM,YM,ZM)
C
C                  GET FIELD COMPONENTS AT XM,YM,ZM
CCC       CALL BTOTAL(XM,YM,ZM,HXM,HYM,HZM)
C
C                  TRANSFORM FIELD COMPONENTS BACK TO DIPOLE SYSTEM
```

```
CCC       HXD = HXM*CSPS - HZM*SNPS
CCC       HYD = HYM
CCC       HZD = HXM*SNPS + HZM*CSPS
C
C                OUTPUT IN SM COORDINATES(GAMMAS)
CCC       BX = - HXD
CCC       BY = - HYD
CCC       BZ = + HZD
C
CCC   END IF
C
C         ALL CALLS WITH ISWIT > 2 GIVE POINT RESULT IN GSM COORDINATES
      IF (ISWIT.GT.2) THEN
C                GET BFIELD AT POINT USING GSM COORDINATE SYSTEM
          XM = - X
          YM = - Y
          ZM = + Z
C
C                    GET BFIELD COMPONENTS AT XM,YM,ZM
          CALL BTOTAL(XM,YM,ZM,HXM,HYM,HZM)
C
C                    OUTPUT IN GSM COORDINATES(GAMMAS)
          BX = - HXM
          BY = - HYM
          BZ = + HZM
C
      END IF
      RETURN
      END
C ***********************************:   COEFFS
      SUBROUTINE COEFFS(NDIM,NELE)
C
C    PURPOSE- CALCULATE COEFFICIENTS
C    INPUT- NDIM, NUMBER OF GRID POINTS FOR NUMERICAL INTEGRATION
C           NELE, NUMBER OF INTERATIONS TO OBTAIN COEFFICIENTS
C
C                FILL COMMON BLOCK SITES WITH BESSEL ROOTS AND VALUES
      CALL ZERO
C
      DO 10  KSA = 1, NELE
C                GET COEFFICIENTS OF LEGENDRE POLYNOMIALS
        CALL SPHRCO(NDIM,KSA)
C                GET COEFFICIENTS OF BESSEL SERIES
        CALL TAILCO(NDIM)
  10  CONTINUE
C
      RETURN
      END
C ***********************************:   CONST
      SUBROUTINE CONST(TI)
C
C    PURPOSE-   PROVIDE CONSTANTS (TO COMMON BLOCKS)
C    INPUT-     TI, DIPOLE TILT ANGLE IN DEGREES
C                 TI>0 INDICATES NORTHERN HEMISPHERE WINTER
C               R, RADIUS OF TAIL (FROM BLOCK 1) IN EARTH RADII
C    OUTPUT-    Q, RATIO OF DIPOLE MOMENT/(RADIUS**3) IN GAMMA
C               PSI, DIPOLE TILT ANGLE IN RADIANS
C               PI, RADIAN EQUIVALENT OF 180 DEGREES
C               SNPS AND CSPS, SIN AND COS OF PSI
C    VARIABLES- DM, DIPOLE MOMENT IN GAMMA*(METERS**3)
C               RE, EARTH RADIUS, METERS
C
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
```

```
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
C
      PI = ACOS(-1.00)
C
      DM = 7.951E+24
      RE = 6.37104E+06
      ZA = R*RE
      Q  = DM/(ZA**3)
C
      PSI = (PI*TI)/180.0
      CSPS = COS(PSI)
      SNPS = SIN(PSI)
C
      RETURN
      END
C ***********************************:   BTOTAL
      SUBROUTINE BTOTAL(XM,YM,ZM,HXM,HYM,HZM)
C
C    PURPOSE-   GET FIELD COMPONENTS USING PREDETERMINED COEFFICIENTS
C    INPUT-     XM,YM,ZM  M COORDINATES
C    OUTPUT-    HXM,HYM,HZM FIELD COMPONENTS IN M SYSTEM IN GAMMAS
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
C
C               TRANSFORM M TO SPHERICAL COORDINATES RR,T,PHI
C               TWO STEP TRANSFORM BECAUSE ZV IS NEEDED BELOW IN 'IF'
      CALL TRANS5(XM,YM,ZM,XV,YV,ZV,D1,D2,D3)
      CALL TRANS3(XV,YV,ZV,RR,T,PHI)
C
      IF(ZV.LT.0.0) THEN
C               GET FIELD FROM DAYSIDE OF SPHERE ZV<0
         CALL DAYTOT(RR,T,PHI,HR,HT,HP)
C                    TRANSFORM TO CARTESIAN ZV DOWN TAIL
         CALL BTRAN1(HR,HT,HP,HXV,HYV,HZV)
      ELSE
C               GET FIELD FROM NIGHTSIDE OF SPHERE ZV>0
         RRZ = SQRT(XV*XV + YV*YV)
         CALL TAILTO(RRZ,PHI,ZV,HR,HP,HZ)
C                    TRANSFORN TO CARTESIAN ZV DOWN TAIL
         CALL BTRAN2(HR,HZ,HP,HXV,HYV,HZV)
      END IF
C
C               TRANSFORM BACK TO M SYSTEM FOR OUTPUT
      HXM = + HZV
      HYM = - HYV
      HZM = + HXV
C
      RETURN
      END
C ***********************************:   TRANS4
      SUBROUTINE TRANS4(XD,YD,ZD,XM,YM,ZM)
C
C    PURPOSE-   TRANSFORM DIPOLE TO M COORDINATES
C    INPUT-     XD,YD,ZD  DIPOLE COORDINATES, ZD NORTH, XD TAILWARD
C    OUTPUT-    XM,YM,ZM  M COORDNATES, XM IS ANTISUNWARD
C    VARIABLES- SNPS AND CSPS, SIN AND COS OF DIPOLE TILT ANGLE
C
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
C
      XM = ZD*SNPS + XD*CSPS
      YM = YD
```

```
      ZM = ZD*CSPS - XD*SNPS
C
      RETURN
      END
C ***********************************:   SPHRCO
      SUBROUTINE SPHRCO(NDIM,KSA)
C
C    PURPOSE-   CALCULATE COEFFICIENTS FOR LEGENDRE POLYNOMIALS
C    INPUT-     NDIM, NUMBER OF GRID POINTS FOR NUMERICAL INTEGRATION
C                     GRID POINTS ARE ON SPHERE SURFACE FROM THETA 0 TO
C                     180 AND PHI=0, THETA=0 POINTS DOWN TAIL
C               KSA, ITERATION LEVEL
C    OUTPUT-    G1,G0, COEFFICIENTS FOR TILT 0 AND 90 DEGREES RESPECT.
C                 CONTAIN ALL EXCEPT ANGULAR AND LEGENDRE FUNCTIONS
C                 SEE EQ. 5.2.3 REF4 FOR GENERAL EXPRESSIONS
C    VARIABLES- Q, RATIO OF DIPOLE MOMENT/RADIUS**3
C               R, RADIUS OF TAIL AND SPHERE
C               B, = R - STANDOFF DISTANCE
C               N, ORDER OF LEGENDRE POLYNOMIAL
C               TI, DIPOLE TILT ANGLE
C               HS, GRID SIZE
C               Y1,Y0, INTEGRAND FOR 0 AND 90 DEGREE CASES
C               Z1,Z0, RESULT OF INTEGRALS FOR 0 AND 90 DEGREE CASES
C
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /CHABLO/ G1(20), G0(20)
      COMMON /BLOCK1/ B,R,Q,PI,PSI
C
C
      IF(KSA.EQ.1) THEN
C               COEFFICIENTS ARE ANALYTIC, 5.11 AND 5.12 REF1
          TI = 0.0
          CALL CONST(TI)
          G1(1) = - 2. * Q
          G0(1) = - 2. * Q
          EP = +1.0
          RRP = 1.0
          RP  = B/R
C
          DO 1  N = 2, 20
            EP = -EP
            RRP = RRP * RP
            G1(N) = - Q * EP * FLOAT(N+1) * RRP
            G0(N) = - Q * EP * FLOAT(N*(N+1)) * RRP
 1        CONTINUE
      ELSE
C               NUMERICAL INTEGRATION REQUIRED FOR KSA>1
C                    GET RADIAL COMPONENT OF DIPOLE PLUS TAIL SOLUTION
C                    ON GRID
          CALL BRDIP(NDIM)
          DO 3  N = 1, 20
C                    FORM INTEGRANDS Y1 AND Y0
             CALL FUN1(N,NDIM)
C                    INTEGRATE, RESULTS Z1 AND Z0
             CALL INTEGR(HS,Y1,Z1,NDIM)
             CALL INTEGR(HS,Y0,Z0,NDIM)
             ZFE = FLOAT(2*N+1)
             FN  = FLOAT(N)
C                    FORM COEFFICIENTS 5.2.3 REF4
             G1(N) = 0.5 * ZFE / (FN*(FN+1.)) * Z1(NDIM)
             G0(N) = 0.5 * ZFE * Z0(NDIM)
 3        CONTINUE
      END IF
C
```

```
      RETURN
      END
C ***********************************:    DAYTOT
      SUBROUTINE DAYTOT(RR,T,PHI,HR,HT,HP)
C
C    PURPOSE-   CALCULATE TOTAL FIELD OF DAYSIDE HALF OF SPHERE ZV<0
C    INPUT-     RR,T,PHI SPHERICAL COORDINATES OF INTEREST
C    OUTPUT-    HR,HT,HP FIELD COMPONENTS IN SPHERICAL SYSTEM
C    VARIABLES- CDIPOL, FRAC. OF DIPOLE PENETRATING OUT OF MAGNETOPAUSE
C               CIMF, FRAC. OF IMF PENETRATING INTO MAGNETOPAUSE
C
      COMMON /BLOCK1/ B, R, Q, PI, PSI
      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
      CD = (1.- CDIPOL)
      CF = (1.- CIMF)
C
      IF(RR.LE.R) THEN
C               GET FIELD INSIDE SPHERE
C                    DIPOLE FIELD
          CALL BDIP (RR,HR1,HT1,HP1)
C                    CHAPMAN-FERRARO DUE TO DIPOLE
          CALL CFDSPH(RR,HR2,HT2,HP2)
C                    TAIL SOLUTION, DAYSIDE INFLUENCE
          CALL BTASPH (RR,HR3,HT3,HP3)
C                    IMF FIELD
          CALL BIMF  (HIMF,XI,D1,D2,D3,D4, HR5,HT5,HP5, D5,D6,D7)
C
C                    EQ. 3.22 INTERNAL FIELD REF1
          HR = HR1  +  CD * (HR2 + HR3)  +  CIMF * HR5
          HT = HT1  +  CD * (HT2 + HT3)  +  CIMF * HT5
          HP = HP1  +  CD * (HP2 + HP3)  +  CIMF * HP5
      ELSE
C               GET FIELD OUTSIDE SPHERE
C                    DIPOLE FIELD
          CALL BDIP (RR,HR1,HT1,HP1)
C                    CHAPMAN-FERRARO DUE TO IMF
          CALL CFISPH(RR,HR2,HT2,HP2)
C                    IMF FIELD
          CALL BIMF  (HIMF,XI,D1,D2,D3,D4, HR5,HT5,HP5, D5,D6,D7)
C
C                    EQ. 3.23 EXTERNAL FIELD REF1
          HR = HR5  +  CF * HR2  +  CDIPOL * HR1
          HT = HT5  +  CF * HT2  +  CDIPOL * HT1
          HP = HP5  +  CF * HP2  +  CDIPOL * HP1
      END IF
      RETURN
      END
C ***********************************:    TAILTO
      SUBROUTINE TAILTO(RR,PHI,Z,HR,HP,HZ)
C
C    PURPOSE-   CALCULATE TOTAL FIELD OF CYLINDER PORTION FOR ZV>0
C    INPUT-     RR,PHI,Z,  CYLINDRICAL COORDINATES
C    OUTPUT-    HR,HP,HZ, FIELD COMPONENTS IN CYLINDRICAL SYSTEM
C    VARIABLES- CDIPOL, FRAC. OF DIPOLE PENETRATING OUT OF MAGNETOPAUSE
C               CIMF, FRAC. OF IMF PENETRATING IN THROUGH MAGNETOPAUSE
C               SNPH,CSPH,  SIN AND COS OF PHI
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
      COMMON /FIMF/   HIMF, CDIPOL, CIMF, XI
C
      CD = (1.- CDIPOL)
```

```
      CF = (1.- CIMF)
C
      IF(RR.LE.R) THEN
C               GET FIELD INSIDE CYLINDER
          RRR = SQRT(RR*RR + Z*Z)
C                    DIPOLE FIELD
          CALL BDIP(RRR,HR1,HT1,HP1)
C                    TRANSFORM SPHERICAL TO CARTESIAN, HZV DOWN TAIL
          CALL BTRAN1(HR1,HT1,HP1, HXV,HYV,HZV)
C                    TRANSFORM CARTEIAN TO CYLINDRICAL
          HR1 = HXV * CSPH  + HYV * SNPH
          HP1 = HYV * CSPH  - HXV * SNPH
          HZ1 = HZV
C                    TOTAL TAIL FIELD SOLUTION
          CALL BTATA(Z,RR,HR2,HZ2,HP2)
C                    CHAPMAN-FERRARO = TOTAL FIELD - DIPOLE FIELD
          HR2 = HR2 - HR1
          HP2 = HP2 - HP1
          HZ2 = HZ2 - HZ1
C                    GET IMF FIELD
          CALL BIMF (HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR5,HZ5,HP5)
C                    SUM OF INTERNAL FIELDS, 3.22 REF1
          HR = HR1  +  CD * HR2  +  CIMF * HR5
          HZ = HZ1  +  CD * HZ2  +  CIMF * HZ5
          HP = HP1  +  CD * HP2  +  CIMF * HP5
      ELSE
C               GET FIELD OUTSIDE OF CYLINDER
          RRR = SQRT(RR*RR + Z*Z)
C                    DIPOLE FIELD
          CALL BDIP(RRR,HR1,HT1,HP1)
C                    TRANSFORM SPHERICAL TO CARTESIAN, HZV DOWN TAIL
          CALL BTRAN1(HR1,HT1,HP1, HXV,HYV,HZV)
C                    TRANSFORM CARTESIAN TO CYLINDRICAL
          HR1 = HXV * CSPH  + HYV * SNPH
          HP1 = HYV * CSPH  - HXV * SNPH
          HZ1 = HZV
C                    CHAPMAN-FERRARO DUE TO IMF
          CALL CFITA(RR,HR2,HZ2,HP2)
C                    IMF FIELD
          CALL BIMF (HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR5,HZ5,HP5)
C                    SUM OF EXTERNAL FIELDS USING 3.23 REF1
          HR = HR5  +  CF * HR2  +  CDIPOL * HR1
          HZ = HZ5  +  CF * HZ2  +  CDIPOL * HZ1
          HP = HP5  +  CF * HP2  +  CDIPOL * HP1
      END IF
      RETURN
      END
C ************************************:    BTRAN1
      SUBROUTINE BTRAN1(HRV,HTV,HPV,HXV,HYV,HZV)
C
C    PURPOSE-   CHANGES FIELD COMPONENTS FROM SPHERICAL TO CARTESIAN
C               HZV POINTS DOWN TAIL AXIS, =0 TO SPHERE CENTER
C    INPUT-     HRV,HTV,HPV, SPHERICAL COMPONENTS
C    OUTPUT-    HXV,HYV,HZV, CARTESIAN COMPONENTS
C    VARIABLES- ST,CT,  SIN AND COS OF THETA
C               CP,SP,  COS AND SIN OF PHI
C
      COMMON /BLOCK5/ CP,SP,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
C
      HXV = HRV*ST*CP + HTV*CT*CP - HPV*SP
      HYV = HRV*ST*SP + HTV*CT*SP + HPV*CP
      HZV = HRV*CT - HTV*ST
C
```

```
      RETURN
      END
C ***********************************:   TAILCO
      SUBROUTINE TAILCO(NDIM)
C
C    PURPOSE-   NUMERICAL CALCULATION OF FOURIER-BESSEL COEFFICIENTS
C    INPUT-     NDIM, NUMBER OF GRID POINTS ALONG CYLINDER CAP AT
C                     THETA=90 DEGREES FOR INTEGRATION
C    OUTPUT-    AX,BX FOURIER-BESSEL COEFFICIENTS WITH MODIFICATIONS
C                     TO BE USEFUL IN FINAL EXPRESSION 6.4 REF1
C    VARIABLES- BE,BO BESSEL FUNCTIONS J1 AND J0 WITH ARGUMENTS THAT
C                     ARE ROOTS OF DERIVATIVES OF THE BESSEL FUNCTIONS
C                     SEE 5.21 AND 5.22 REF1
C               Z1,Z0 RESULTS OF INTEGRATION FOR 0 AND 90 DEGREES CASES
C               ZE,ZO ZEROS OF DERIVATIVES OF J1 AND J0
C               Q,    DIPOLE MOMENT/(RADIUS**3)
C
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK1/ B ,R, Q, PI, PSI
      COMMON /BLOCK2/ ZE(20), AX(20), ZO(20), BX(20)
      COMMON /BLOCK7/ BE(20), BO(20)
C
C               GENERATE POTENTIAL ON CYLINDER HEAD= CHAP-FERR + DIPOLE
      CALL TAPOT(NDIM)
C               GENERATE COEFFICIENTS FOR N ORDERS
      DO 1  N = 1, 20
C                    FORM INTEGRANDS Y1 AND Y0
          CALL FUN2(N,NDIM)
C                    DO INTEGRATIONS FOR BOTH TILT CASES,RESULT Z1, Z0
          CALL INTEGR(HS,Y1,Z1,NDIM)
          CALL INTEGR(HS,Y0,Z0,NDIM)
C                    PREPARE SOME BESSEL FACTORS
          BX0 = BE(N) * BE(N)
          BY0 = BO(N) * BO(N)
          ZEX = ZE(N) * ZE(N)
C                    COEFFICIENTS A AND B OF 5.24 AND 5.25 REF1
          AX(N) = Z1(NDIM) * ZEX / (BX0 * (ZEX-1.))
          BX(N) = Z0(NDIM) / (BY0)
C                    MODIFY COEFFICIENTS TO SUIT EXPRESSION 6.4 REF1
          AX(N) = AX(N) * ZE(N) * Q
          BX(N) = BX(N) * ZO(N) * 2. * Q
    1   CONTINUE
C
      RETURN
      END
C ***********************************:   TRANS3
      SUBROUTINE TRANS3(XV,YV,ZV,RR,T,PHI)
C
C    PURPOSE-   CHANGE CARTESIAN TO SPHERICAL AND FORM TRIG. FUNCTIONS
C               FOR COMMON BLOCKS
C    INPUT-     XV,YV,ZV  CARTESIAN WITH ZV DOWN TAIL
C    OUTPUT-    RR,T,PHI  SPHERICAL COORDINATES
C    VARIABLES- Q, RATIO DIPOLE MOMENT/(RADIUS**3)
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
C
      RR  = SQRT(XV*XV + YV*YV + ZV*ZV)
      PHI = ATAN3(YV,XV)
      T = ATAN3(SQRT(XV*XV+YV*YV),ZV)
C
      CT = COS(T)
      ST = SIN(T)
```

```
      CSPH = COS(PHI)
      SNPH = SIN(PHI)
      QCS = Q*CSPH
      QSN = Q*SNPH
C
      RETURN
      END
C ***********************************:    TRANS5
      SUBROUTINE TRANS5(XM,YM,ZM, XV,YV,ZV, XGSE,YGSE,ZGSE)
C
C    PURPOSE-    TRANSFORM M TO SPHERICAL AND GSE SYSTEM
C    INPUT-      XM,YM,ZM THE M SYSTEM, EARTH CENTERED, XM DOWN TAIL
C    OUTPUT-     XV,YV,ZV CARTESIAN, SPHERE CENTERED, ZV DOWN TAIL
C                XGSE,YGSE,ZGSE, GSE SYSTEM
C    VARIABLES- BEX, = RADIUS - STAND-OFF DISTANCE
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      XV = ZM
      YV = -YM
      ZV = XM - BEX
C
      XGSE = - XM
      YGSE = -YM
      ZGSE = + ZM
C
      RETURN
      END
C ***********************************:    ZERO
      SUBROUTINE ZERO
C
C    PURPOSE-    FILL COMMON BLOCKS WITH ZEROS OF BESSELS AND FUNCTION
C                VALUES AT ZEROS
C    INPUT-      SOURCE IS TABLE 9.5 REF2
C                ZE,ZEROS OF J1'(XI)
C                ZO, ZEROS OF JO'(YI)
C                BE, BESSEL VALUE J1(XI)
C                BO, BESEL VALUE JO(YI)
C
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK7/ BE(20),BO(20)
C
C
C                ZEROS    J1'(XI) = 0.0
C
      ZE(1)= 1.84118
      ZE(2)= 5.33144
      ZE(3)= 8.53632
      ZE(4)= 11.70600
      ZE(5)= 14.86359
      ZE(6)= 18.01553
      ZE(7)= 21.16437
      ZE(8)= 24.31133
      ZE(9)= 27.45705
      ZE(10)= 30.60192
      ZE(11)= 33.74618
      ZE(12)= 36.88999
      ZE(13)= 40.03344
      ZE(14)= 43.17663
      ZE(15)= 46.31960
      ZE(16)= 49.46239
      ZE(17)= 52.60504
      ZE(18)= 55.74757
      ZE(19)= 58.89000
      ZE(20)= 62.03235
```

```
C
C
C          ZEROS      J0'(YI) = 0.0
C
      ZO(1) = 3.8317059702
      ZO(2) = 7.0155866698
      ZO(3) = 10.1734681351
      ZO(4) = 13.3236919363
      ZO(5) = 16.4706300509
      ZO(6) = 19.6158585105
      ZO(7) = 22.7600843806
      ZO(8) = 25.9036720876
      ZO(9)  = 29.0468285349
      ZO(10) = 32.1896799110
      ZO(11) = 35.3323075501
      ZO(12) = 38.4747662348
      ZO(13) = 41.6170942128
      ZO(14) = 44.7593189977
      ZO(15) = 47.9014608872
      ZO(16) = 51.0435351836
      ZO(17) = 54.1855536411
      ZO(18) = 57.3275254379
      ZO(19) = 60.4694578453
      ZO(20) = 63.61141
C
C
C         BE(I) = J1(XI)     BESSELS EVALUATED AT XI'S OF ABOVE
C
      BE(1)= +0.58187
      BE(2)= -0.34613
      BE(3)= +0.27330
      BE(4)= -0.23330
      BE(5)= +0.20701
      BE(6)= -0.18802
      BE(7)= +0.17346
      BE(8)= -0.16184
      BE(9)= +0.15228
      BE(10)= -0.14424
      BE(11)= +0.13736
      BE(12)= -0.13137
      BE(13)= +0.12611
      BE(14)= -0.12143
      BE(15)= +0.11724
      BE(16)= -0.11345
      BE(17)= +0.11001
      BE(18)= -0.10687
      BE(19)= +0.10397
      BE(20)= -0.10131
C
C
C         BO(I) = J0(YI)     BESSELS EVALUATED AT YI'S OF ABOVE
C
      BO(1) = -0.4027593957
      BO(2) = +0.3001157525
      BO(3) = -0.2497048871
      BO(4) = +0.2183594072
      BO(5) = -0.1964653715
      BO(6) = +0.1800633753
      BO(7) = -0.1671846005
      BO(8) = +0.1567249836
      BO(9)  = -0.1480111100
      BO(10) = +0.1406057982
      BO(11) = -0.1342112403
      BO(12) = +0.1286166221
```

```
      BO(13) = -0.1236679608
      BO(14) = +0.1192498120
      BO(15) = -0.1152736941
      BO(16) = +0.1116704969
      BO(17) = -0.1083853489
      BO(18) = +0.1053740554
      BO(19) = -0.1026005671
      BO(20) = +0.1000
C
C
      RETURN
      END
C ***********************************:   BIMF
      SUBROUTINE BIMF(HIMF,XI, HXM,HZM, HXV,HZV, HR,HT,HP, HRHO,HZ,HPHI)
C
C    PURPOSE-   DETERMINE IMF CONTRIBUTION IN CARTESIAN M AND V,SPHER,
C               AND CYLINDRICAL COORDINATES
C               NO Y COMPONENT INCLUDED
C    INPUT-     HIMF, MAGNITUDE OF IMF (GAMMA)
C               XI, DIRECTION OF IMF, 0 DEGREES IS NORTHWARD, 90 ANTISUN
C    OUTPUT-    HXM,HZM  FIELD VALUES IN M SYSTEM, HXM DOWN TAIL
C               HXV,HZV  FIELD VALUES IN V SYSTEM, HZV DOWN TAIL
C               HR,HT,HP   "     "    "  SPHERICAL
C               HRHO,HZ,HPHI     "    "  CYLINDRICAL
C    VARIABLES- ST,CT  SIN AND COS OF THETA
C               SNPH,CSPH  SIN AND COS OF PHI
C
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
C               DEGREES TO RADIANS
      AXI = XI*3.1415926535/180.0
C               M SYSTEM
      HXM = HIMF * SIN(AXI)
      HZM = HIMF * COS(AXI)
C               V SYSTEM
      HXV = HZM
      HZV = HXM
C               SPHERICAL SYSTEM
      HR = + HXV*ST*CSPH + HZV*CT
      HT = + HXV*CT*CSPH - HZV*ST
      HP = - HXV*SNPH
C               CYLINDRICAL SYSTEM
      HRHO = + HXV * CSPH
      HPHI = + HP
      HZ   = + HZV
C
      RETURN
      END
C ***********************************:   CFDSPH
      SUBROUTINE CFDSPH(RR,HR,HT,HP)
C
C    PURPOSE-   CALCULATE FIELD OF CHAPMAN-FERRARO CURRENTS FOR
C               SHIELDING DIPOLE ONLY IN SPHERE. SUMMATION OF FIELD DUE
C               TO POTENTIAL 5.11 REF1
C               NOTE: RECURRENCE RELATIONS JAHNKE-EMDE NORMALIZED, REF3
C    INPUT-     RR, RADIAL LOCATION
C    OUTPUT-    HR,HT,HP SPHERICAL FIELD COMPONENTS, 5.2.13 REF4
C    VARIABLES- GAM1,GAMO  LEGENDRE POLYNOMIAL COEFFICIENTS FROM SPHRCO
C               PN, LEGENDRE POLYNOMIAL M=0
C               CPN, DERIVATIVE OF PN W.R.T. ARGUMENT COS(THETA)
C               TPN, DERIVATIVE OF PN W.R.T. THETA
C               PNM, LEGENDRE POLYNOMIAL M=1
C               TPNM, DERIVATIVE OF PNM W.R.T. THETA
C               SPNM, QUOTIENT PN/SIN(THETA)
```

```
C               S1,C1 SIN AND COS OF THETA
C
      DIMENSION  PN(20), CPN(20), TPN(20)
      DIMENSION PNM(20),TPNM(20),SPNM(20)
      COMMON /CHABLO/ GAM1(20), GAM0(20)
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ C1, S1
C
      D = 1.0E-03
C
      C2 = C1*C1 - S1*S1
      S2 = 2.*S1*C1
C
      PN(1) = C1
      PN(2) = 0.25*(3.*C2+1.)
      CPN(1) = 1.0
      CPN(2) = 3.*C1
C
      DO 1  N=3,20
          FN = FLOAT(N)
          ZFE = 2.*FN - 1.0
          PN(N)  = (1./FN) * (ZFE*C1*PN(N-1) - (FN-1.)*PN(N-2))
          CPN(N) = ZFE*PN(N-1) + CPN(N-2)
    1 CONTINUE
C
      DO 2  N=1,20
          FN = FLOAT(N)
          TPN(N)   = - S1*CPN(N)
          PNM(N)   = - TPN(N)
          SPNM(N)  = + CPN(N)
          TPNM(N)  = FN*(FN+1.)*PN(N)  - C1*CPN(N)
    2 CONTINUE
C
      HR1 = GAM1(1)  * PNM(1)
      HR2 = GAM0(1)  * PN(1)
      HT1 = GAM1(1)  * TPNM(1)
      HT2 = GAM0(1)  * TPN(1)
      HP  = GAM1(1)  * SPNM(1)
C
      RP = 0.0
      IF(RR.GT.D)  RP = (RR/R)
      RRP = 1.0
C
      DO 3  N=2,20
          FN = FLOAT(N)
C
          RRP = RRP * RP
C
          GAMA = GAM1(N)  * RRP
          GAMB = GAM0(N)  * RRP
C
C              1 AND 2 INDICATE TERMS MEANT FOR 0 AND 90 DEGREE CASES
          HR1 = HR1 + GAMA * PNM(N)
          HR2 = HR2 + GAMB * PN(N)
C
          HT1 = HT1 + GAMA * TPNM(N)/ FN
          HT2 = HT2 + GAMB * TPN(N)  / FN
C
          HP  = HP  + GAMA * SPNM(N)/ FN
C
    3 CONTINUE
C
```

```
C               EACH PART RELATING TO TILT CASES IS COMBINED WITH
C               FUNCTIONS OF 5.9 REF1
      HR = - (HR1 * CSPH * CSPS  +  HR2 * SNPS)
      HT = - (HT1 * CSPH * CSPS  +  HT2 * SNPS)
      HP = +  HP   * SNPH * CSPS
C
C
      RETURN
      END
C ***********************************:   CFISPH
      SUBROUTINE CFISPH(RR,HR,HT,HP)
C
C    PURPOSE-   CALCULATE CHAPMAN-FERRARO FIELD OUTSIDE SHERE DUE TO IMF
C               USING 5.15 REF1
C    INPUT-     RR, RADIAL COORDINATE
C    OUTPUT-    HR,HT,HP CHAP-FERR FIELD COMPONENTS IN SPHERICAL SYSTEM
C    VARIABLES- HR3,HT3,HP3  IMF FIELD COMPONENTS FROM SUBROUT. BIMF
C
C
      COMMON /BLOCK1/ BEX, R, Q, PI, PSI
      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
C               GET IMF FIELD COMPONENTS
      CALL BIMF(HIMF,XI, D1,D2,D3,D4, HR3,HT3,HP3, D5,D6,D7)
C               TRANSFORM COMPONENTS TO SPHERICAL SYSTEM
      FR = R/RR
      FR = FR*FR*FR
C
      HR = - HR3 * FR
      HT = + 0.5 * HT3 * FR
      HP = + 0.5 * HP3 * FR
C
      RETURN
      END
C ***********************************:   BDIP
      SUBROUTINE BDIP(RR,HR,HT,HP)
C
C    PURPOSE-   CALCULATE EXPLICIT COMPONENTS OF DIPOLE USING SPHERICAL
C               EQUIVALENT OF 5.6 REF1 FOR POTENTIAL
C    INPUT-     RR, RADIAL COORDINATE LOCATION
C    OUTPUT-    HR,HT,HP DIPOLE FIELD IN SPHERICAL SYSTEM   (GAMMA)
C    VARIABLES- C1,S1 COS AND SIN OF THETA
C               R, RADIUS OF SPHERE AND TAIL
C               B, = R - STANDOFF DISTANCE
C               SNPS,CSPS SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ C1, S1
C
      R3 = R**3
C
      A = QCS*R3
      AA = QSN*R3
C
      V = ((RR*RR) + (B*B) + (2.*B*RR*C1))
C
      AV = ABS(V)
      SV = SIGN(1.0,V)
C
      W3 = SQRT(AV**3)
      W4 = SQRT(AV**5)
C
      W1 = W3 * SV
```

```
      W2 = W4 * SV
C
      F1 = 2.*RR*B*S1
      G1 = 2.*RR + 2.*B*C1
      E1 = 1.5*RR*G1/W2
      E2 = 1.5*F1/W2
C
      HR1 = A*CSPS*S1*(1./W1 - E1)
      HR2 = Q*R3*C1*SNPS*(1./W1 - E1)
      HR3 = -Q*R3*B*SNPS*3.*(RR+ B*C1)/W2
C
      HT1 = A*CSPS*(C1/W1 + S1*E2)
      HT2 = Q*R3*SNPS*(C1*E2 - S1/W1)
      HT3 = Q*R3*B*SNPS*(3.*B*S1/W2)
C
      HP = -AA*CSPS/W1
C
      HR = HR1 + HR2 + HR3
      HT = HT1 + HT2 + HT3
      HP = HP
C
      RETURN
      END
C ***********************************:   FUN1
      SUBROUTINE FUN1(N,NDIM)
C
C    PURPOSE-   FORM INTEGRAND FOR COEFFICIENT CALCULATIONS AT EACH OF
C               NDIM STEPS FOR A CERTAIN ORDER OF LEGENDRE POLYNOMIAL
C    INPUT-     N, ORDER OF LEGENDRE POLYNOMIAL
C               NDIM, NUMBER OF GRID POINTS USED FOR INTEGRATION
C    OUTPUT-    Y1,YO VALUE OF INTEGRAND FOR CASES 0 AND 90 DEGREES RESP
C    VARIABLES- PNM, LEGENDRE POLYNOMIAL M=1
C               PN,  LEGENDRE POLYNOMIAL M=0
C               ST, SIN(T) LAST VALUE COMES FROM SUBROUT. LEGEN
C
      COMMON /LEG1/ PN(20), PNM(20), CPN(20)
      COMMON /BLOCK8/ W1(1001),W0(1001)
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK6/ CT, ST
C
C
      DO 1  I = 1, NDIM
C               X(I) IS THETA, GENERATES LEGENDRE POLYNOMIAL TERMS
         CALL LEGEN(X(I),N)
C
         D = 1.0E-20
C
         IF(ABS(W1(I)).LT.D)   W1(I) = 0.0
         IF(ABS(W0(I)).LT.D)   W0(I) = 0.0
         IF (ABS(PNM(N)).LT.D)    PNM(N) = 0.0
         IF(ABS(PN(N)).LT.D)     PN(N)  = 0.0
         IF(ABS(ST).LT.D)        ST = 0.0
C
C               NDIM INTEGRANDS FOR EACH ORDER TO GET COEFFICIENTS
C               5.2.3 REF4
         Y1(I) = W1(I) * PNM(N) * ST
         Y0(I) = W0(I) * PN(N)  * ST
C
   1  CONTINUE
C
C
      RETURN
      END
C ***********************************:   BRDIP
```

```
      SUBROUTINE BRDIP(NDIM)
C
C     PURPOSE-   GENERATE RADIAL COMPONENTS OF DIPOLE FIELD ON GRID FOR
C                TILT 0 AND 90 DEGREES
C     INPUT-     NDIM, NUMBER OF GRID POINTS AT RADIUS=R ALONG CURVE
C                THETA=0 TO 180 DEGREES TO BE USED FOR INTEGRATION
C     OUTPUT-    W1,WO  RADIAL COMPONENTS OF DIPOLE, INTO BLOCK 8
C     VARIABLES- KDIM, NUMBER OF THE GRID POINT SEPARATING DAY AND NIGHT-
C                      SIDE OF SPHERE
C                R, RADIUS OF SHPERE AND TAIL
C                Q, RATIO DIPOLE MOMENT/(R**3)
C                SNPH,CSPH SIN AND COS OF PHI
C                QSN,QCS    =Q*SNPH AND =Q*CSPH RESPECTIVELY
C                X(I), THETA VALUES OF GRID POINTS
C                HS, GRID POINT SPACING
C
      COMMON /BLOCK8/ W1(1001),W0(1001)
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT,ST
      COMMON /BLOCK3/ H0,ALFA,KSWIT
C
      KDIM = (NDIM-1)/2
C
C                PHI = 0 DEGREES FOR THIS SUBROUT.
      CSPH = 1.0
      SNPH = 0.0
      QCS = Q
      QSN = 0.0
C
C
C                GENERATE GRID POINTS
      X(1) = 0.0
      HS = PI/FLOAT(NDIM-1)
      DO 1  I = 2, NDIM
          X(I) = FLOAT(I-1) * HS
    1 CONTINUE
C
C                GET COMPONENTS FOR TILT=0 DEGREES
      TI = 0.0
      CALL CONST(TI)
C
      DO 4  I = 1, NDIM
C
          CT = COS(X(I))
          ST = SIN(X(I))
C
          RRDIP = R
C                GET RADIAL COMPONENT OF DIPOLE AT SPHERE RADIUS
          CALL BDIP(RRDIP,HR,HT,HP)
          W1(I) = HR
C
          IF (I.LE.KDIM) THEN
C                     LOCATION IS NIGHTSIDE, CHANGE TO CYLINDRICAL
              RHO= R * ST
              Z  = R * CT
C                     GET CYLINDER SOLUTION
              CALL VACTA(Z,RHO,HR,HZ,HP)
              HR = HR * ST + HZ * CT
C                     CYLINDER SOLUTION HAS DIPOLE + CHAP-FERR 5.16 REF1
              W1(I) = W1(I) - HR
          ELSE
C                     LOCATION IS DAYSIDE OF SPHERE
```

```
            RRSHE = R
C                   DAYSIDE FIELD RESULTING FROM TAIL SOLUTION
            CALL BTASPH(RRSHE,HR,HT,HP)
C                   DIPOLE PLUS DAYSIDE TAIL SOLUTION GIVES DAYSIDE
C                   FIELD, 5.1 REF1
            W1(I) = W1(I) + HR
         END IF
  4   CONTINUE
C
C
C               GENERATE COMPONENTS FOR TILT=90 DEGREES
C               (SAME PROCEDURE AS ABOVE FOR TILT=0 DEGREES)
      TI = 90.0
      CALL CONST(TI)
C
C
      DO 7  I = 1, NDIM
C
         CT = COS(X(I))
         ST = SIN(X(I))
C
         RRDIP = R
         CALL BDIP(RRDIP,HR,HT,HP)
         W0(I) = HR
C
         IF (I.LE.KDIM) THEN
            RHO= R * ST
            Z  = R * CT
            CALL VACTA(Z,RHO,HR,HZ,HP)
            HR = HR * ST + HZ * CT
            W0(I) = W0(I) - HR
         ELSE
            RRSHE = R
            CALL BTASPH(RRSHE,HR,HT,HP)
            W0(I) = W0(I) + HR
         END IF
   7  CONTINUE
C
C
      RETURN
      END
C ************************************:   INTEGR
      SUBROUTINE INTEGR(H,Y,Z,NDIM)
C
C    PURPOSE-   GENERAL INTEGRATION ROUTINE
C    INPUT-     H, GRID SIZE
C               Y, INTEGRAND, FROM FUN1 OR FUN2 SUBROUT.
C               NDIM, NUMBER OF GRID POINTS USED
C    OUTPUT-    Z INTEGRAL RESULT OF FUNCTION Y FROM ZERO TO NDIM*H
C               (USED TO FILL Z1 AND ZO OF BLOCK 9)
C
      DIMENSION Y(NDIM), Z(NDIM)
C
      HT=.3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
```

```
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
      Z(1)=0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
    2 DO 4 I=7,NDIM,2
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
C     END OF INTEGRATION LOOP
C
    7 IF(NDIM-3)12,11,8
C
C     NDIM IS EQUAL TO 4 OR 5
    8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
      SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      Z(1)=0.
      AUX1=Y(3)+Y(3)
      AUX1=AUX1+AUX1
      Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
      IF(NDIM-5)10,9,9
    9 AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
   10 Z(3)=SUM1
      Z(4)=SUM2
      RETURN
C
C     NDIM IS EQUAL TO 3
   11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
      SUM2=Y(2)+Y(2)
      SUM2=SUM2+SUM2
      Z(3)=HT*(Y(1)+SUM2+Y(3))
      Z(1)=0.
      Z(2)=SUM1
   12 RETURN
      END
C ***********************************:   BTRAN2
      SUBROUTINE BTRAN2(HRV,HZV,HPV,HXV,HYV,HZVV)
C
C    PURPOSE-   CHANGE FIELD COMPONENTS FROM CYLINDRICAL TO CARTESIAN
```

```
C    INPUT-     HRV,HZV,HPV  CYLINDRICAL FIELD COMPONENTS, HZV DOWN TAIL
C    OUTPUT-    HXV,HYV,HZVV  CARTESIAN COMPONENTS, HZVV DOWN TAIL
C    VARIABLES- SP,CP SIN AND COS OF PHI
C
      COMMON /BLOCK5/ CP,SP,QCS,QSN,CSPS,SNPS
C
      HXV = HRV*CP - HPV*SP
      HYV = HRV*SP + HPV*CP
      HZVV = HZV
C
      RETURN
      END
C ***********************************:   BTASPH
      SUBROUTINE BTASPH(RR,HR,HT,HP)
C
C   PURPOSE-    CALCULATE FIELD COMPONENTS OF TAILFIELD WITHIN SPHERE
C               FOR ZV<0. (IMAGE OF VACTA) COMPONENTS DERIVED FROM 5.26
C               REF1 WITH SIGN CHANGE IN EXPONENT DUE TO ZV POSITION
C               FIRST CALLED FROM SPHRCO ON ITERATION 2. 15 TERMS USED.
C    INPUT-     RR, RADIAL DISTANCE IN SPHERICAL COORDINATES
C    OUTPUT-    HR,HT,HP SPHERICAL FIELD COMPONENTS
C    VARIABLES- RHO,Z CYLINDRICAL EQUIVALENTS OF INPUT
C               ST,CT SIN AND COS OF THETA
C               ZE, ZEROS OF J1'( ) BESSEL       5.23 REF1
C               ZO, ZEROS OF JO'( ) BESSEL         "
C               AX,BX FOURIER-BESSEL COEFFICIENTS WITH ADDITIONS
C               ALPHA, PLASMA CONTENT OF TAIL 1 MEANS VACUUM
C                                           0 MEANS HARRIS SHEET
C                   ALPHA=1 USED TO GET COEFFICIENTS
C                   ALPHA NOT = 1 ONLY POSSIBLE FOR BTOTAL APPLICATION
C
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT,ST
      COMMON /BLOCK3/ H0,ALFA,KSWIT
C
C               CHANGE INPUT TO CYLINDRICAL COORDINATES
      RHO = RR * ST
        Z = RR * CT
C
      FL = 0.5 * (1.-ALFA)
C
      HHR = 0.0
      HHZ = 0.0
      HHP = 0.0
      THR = 0.0
      THZ = 0.0
      D = 1.0E-02
C
C
      DO 100 I = 1,15
C
          ARX = ZE(I)/R * RHO
          ARZ = ZE(I)/R * Z
          TARX = ZO(I)/R * RHO
          TARZ = ZO(I)/R * Z
C
          EX = EXP(+ARZ)
          EX = EX * FL * AX(I)
          TEX = EXP(+TARZ)
          TEX = TEX * FL * BX(I)
C
```

```
          B0  = 1.0
          B1  = 0.0
          B2  = 0.0
          TB0 = 1.0
          TB1 = 0.0
          IF(ABS(RHO).GE.1.0E-06) THEN
              B0  = BFJ0(ARX)
              B1  = BFJ1(ARX)
              B2  = BFJ2(B1,B0,ARX)
              TB0 = BFJ0(TARX)
              TB1 = BFJ1(TARX)
          END IF
              HHR = HHR + (B0-B2)*EX
              HHP = HHP + (B0+B2)*EX
              HHZ = HHZ + 2.0*B1*EX
C
              THR = THR + TB1*TEX
              THZ = THZ + TB0*TEX
C
C
 100  CONTINUE
C
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS* SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C
      HHR = - HHR + THR
      HHP = + HHP
      HHZ = - HHZ - THZ
C               OUTPUT IN SPHERICAL COORDINATES
      HR = HHR*ST + HHZ*CT
      HT = HHR*CT - HHZ*ST
      HP = HHP
C
      RETURN
      END
C ***********************************:   ATAN3
      FUNCTION ATAN3(Y,X)
C
      PI = 3.1415926535
      ATAN3 = 0.00
      IF((X.NE.0.0).OR.(Y.NE.0.0))   ATAN3 = ATAN2(Y,X)
C
      RETURN
      END
C ***********************************:   BFJ0
      FUNCTION BFJ0(X)
C
C    PURPOSE-   CALCULATE BESSEL FUNCTION J0(X)
C               USING 9.4.1 AND 9.4.3 OF REF2 TO ORDER 12 APPROXIMATION
C    INPUT-     X, ARGUMENT OF THE BESSEL FUNCTION
C    OUTPUT-    BFJ0(X), =J0(X)
C
      DATA A1/-2.2499997/,A2/1.2656208/,A3/-0.3163866/,A4/0.0444479/,
     1 A5/-0.0039444/,A6/0.00021/,B0/0.79788456/,B1/-0.00000077/,
     2 B2/-0.0055274/,B3/-0.00009512/,B4/0.00137237/,B5/-0.00072805/,
     3 B6/0.00014476/,C0/-0.78539816/,C1/-0.04166397/,C2/-0.00003954/,
     4 C3/0.00262573/,C4/-0.00054125/,C5/-0.00029333/,C6/0.00013558/
C
      X0=ABS(X)
      T=X0/3.0
```

```
C
      IF (T.LT.1.0) THEN
C               USE EQ. 9.4.1
          T=T*T
          H1=1.0+T*(A1+T*(A2+T*(A3+T*(A4+T*(A5+T*A6)))))
          BFJ0=H1
      ELSE
C               USE EQ. 9.4.3
          T=1.0/T
          H2 =B0+T*(B1+T*(B2+T*(B3+T*(B4+T*(B5+T*B6)))))
          H3=X0+C0+T*(C1+T*(C2+T*(C3+T*(C4+T*(C5+T*C6)))))
          BFJ0=H2*COS(H3)/SQRT(X0)
      END IF
      RETURN
      END
C ***********************************:    BFJ1
      FUNCTION BFJ1(X)
C
C    PURPOSE-   CALCULATE BESSEL FUNCTION J1(X)
C               USING 9.4.4 AND 9.4.6 REF2 TO ORDER 12 APPROXIMATION
C    INPUT-     X, ARGUMENT OF BESSEL FUNCTION
C    OUTPUT-    BFJ1, BESSEL FUNCTION J1(X)
C
      DATA A1/-.56249985/,
     1 A2/.21093573/,A3/-.03954289/,A4/.00443319/,A5/-.00031761/,
     2 A6/.00001109/,B0/.79788456/,B1/.00000156/,B2/.01659667/,
     3 B3/.00017105/,B4/-.00249511/,B5/.00113653/,B6/-.00020033/,
     4 C0/-2.35619449/,C1/.12499612/,C2/.0000565/,C3/-.00637879/,
     5 C4/.00074348/,C5/.00079824/,C6/-.00029166/
C
      X0=ABS(X)
      X1=SIGN(1.0,X)
      T=X0/3.0
C
      IF (T.LT.1.0) THEN
C               USE 9.4.4
          T=T*T
          H1=0.5+T*(A1+T*(A2+T*(A3+T*(A4+T*(A5+T*A6)))))
          BFJ1=X*H1
      ELSE
C               USE 9.4.6
          T=1.0/T
          H2=B0+T*(B1+T*(B2+T*(B3+T*(B4+T*(B5+T*B6)))))
          H3=X0+C0+T*(C1+T*(C2+T*(C3+T*(C4+T*(C5+T*C6)))))
          BFJ1=X1*H2*COS(H3)/SQRT(X0)
      END IF
      RETURN
      END
C ***********************************:    BFJ2
      FUNCTION BFJ2(B1,B0,X)
C
C    PURPOSE-   CALCULATE BESSEL FUNCTION J2(X) USING 9.1.27 REF2
C    INPUT-     B1, J1(X) BESSEL FROM FUNCTION BFJ1
C               B0, J0(X) BESSEL FROM FUNCTION BFJ0
C               X, ARGUMENT OF BESSEL FUNCTIONS
C    OUTPUT-    BFJ2, BESSEL J2(X)
C
      BFJ2 = 0.0
      IF(X.GT.0.0)  THEN
          BFJ2 = (2./X) * B1 - B0
      END IF
C
      RETURN
      END
```

```
C *************************************:    CFITA
      SUBROUTINE CFITA(RR,HR,HZ,HP)
C
C     PURPOSE-   CALCULATE CHAPMAN-FERRARO FIELD OF IMF OUTSIDE CYLINDER
C     INPUT-     RR  RHO IN CYLINDRICAL COORDINATES
C     OUTPUT-    HR,HZ,HP CYLINDRICAL FIELD COMPONENTS
C     VARIABLES- HIMF, STRENGTH OF IMF FIELD
C                XI, IMF ANGLE
C                R, RADIUS OF MAGNETOPAUSE SPHERE AND TAIL
C
      COMMON /BLOCK1/ BEX, R, Q, PI, PSI
      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
C                       GET IMF FIELD COMPONENTS
      CALL BIMF(HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR3,HZ3,HP3)
C
      FR = R/RR
      FR = FR*FR*FR
C
C                       COMPONENTS DERIVED FROM 5.30 REF1
      HR = - HR3 * FR
      HZ = + 0.5 * HZ3 * FR
      HP = + 0.5 * HP3 * FR
C
      RETURN
      END
C *************************************:    FUN2
      SUBROUTINE FUN2(N,NDIM)
C
C     PURPOSE-   CALCULATE INTEGRAND FOR FOURIER-BESSEL INTEGRALS. ONE
C                FOR EACH OF NDIM GRID POINTS. DONE FOR TILT 0 AND 90 DEG
C     INPUT-     N, ORDER OF BESSEL FUNCTION
C                NDIM, NUMBER OF GRID POINTS
C     OUTPUT-    Y1,Y0 INTEGRANDS FOR TILT 0, 90 DEGREES, 5.24,5.25 REF1
C                OUTPUT TO BLOCK 9
C     VARIABLES- ZE, ZEROS OF DERIVATIVES OF BESSEL J1
C                ZO,   "     "        "       "     "    J0
C                X, RADIAL VALUES OF GRID POINTS
C                R, RADIUS OF TAIL CYLINDER
C
      COMMON /BLOCK8/ W1(1001),W0(1001)
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK1/ B ,R, Q, PI, PSI
      COMMON /BLOCK2/ ZE(20), AX(20), ZO(20), BX(20)
C
C
      DO 1  I = 1, NDIM
C                       ARGUMENTS OF BESSELS
          ARX  = ZE(N) * X(I) / R
          TARX = ZO(N) * X(I) / R
C                       SET BESSEL VALUES FOR VERY SMALL ARGUMENTS
          B1 = 0.0
          B0 = 1.0
          IF(ABS(X(I)).GE.1.0E-06) THEN
C                       GET BESSEL FUNCTION VALUES J1 AND J0
              B1 = BFJ1(ARX)
              B0 = BFJ0(TARX)
          END IF
C                       INTEGRANDS OF 5.24 AND 5.25 REF1
              Y1(I) = X(I) * W1(I) * B1
              Y0(I) = X(I) * W0(I) * B0
    1 CONTINUE
C
C
```

```
      RETURN
      END
C ***********************************:    LEGEN
      SUBROUTINE LEGEN(T,NE)
C
C    PURPOSE-   CALCULATE LEGENDRE POLYNOMIAL TERMS
C    INPUT-     T, THETA OF SPHERICAL SYSTEM
C               NE, ORDER OF LEGENDRE POLYNOMIAL
C    OUTPUT-    PN, LEGENDRE POLYNOMIAL TERMS M=0
C               CPN, DERIVATIVE OF PN W.R.T. COS(T)
C               PNM, LEGENDRE POLYNOMIAL TERMS M=1
C
      COMMON /LEG1/ PN(20), PNM(20), CPN(20)
      COMMON /BLOCK6/ C1, S1
C
      C1 = COS(T)
      S1 = SIN(T)
C
      C2 = C1*C1 - S1*S1
      S2 = 2.*S1*C1
C
      PN(1) = C1
      PN(2) = 0.25*(3.*C2+1.)
C
      CPN(1) = 1.0
      CPN(2) = 3.*C1
C
      PNM(1) = + S1 * CPN(1)
      PNM(2) = + S1 * CPN(2)
C
C
      IF(NE.LT.3)   RETURN
      IF(NE.EQ.3)   NELE= 4
      IF(NE.GT.3)   NELE= NE
C
      DO 1  N=3,NELE
          FN = FLOAT(N)
          ZFE = 2.*FN - 1.0
C               RECURRENCE RELATION FROM REF3 JAHNKE-EMDE NORMALIZED
          PN(N)  = (1./FN) * (ZFE*C1*PN(N-1) - (FN-1.)*PN(N-2))
          CPN(N) = ZFE*PN(N-1) + CPN(N-2)
          PNM(N)  = + S1*CPN(N)
  1   CONTINUE
C
C
      RETURN
      END
C ***********************************:    BTATA
      SUBROUTINE BTATA(Z,RHO,HHR,HHZ,HHP)
C
C    PURPOSE-   CALCULATE COMPONENTS OF TAILFIELD SOLUTION  FOR ZV>0
C               IINCLUDING STRETCH,SUMMARIES 6.3 AND 6.4 REF1
C    INPUT-     Z,RHO CYLINDRICAL COORDINATES
C    OUTPUT-    HHR,HHZ,HHP CYLINDRICAL FIELD COMPONENTS
C    VARIABLES- ALFA, PLASMA CONTENT STRETCH PARAMETER
C               SNPH,CSPH  SIN AND COS OF PHI, A CYLINDRICAL COORDINATE
C               SNPS,CSPS  SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C               AX,BX  COEFFICIENTS FOR FOURIER-BESSEL SERIES
C               R, RADIUS OF TAIL
C               ZE,ZO  ZEROS OF BESSEL FUNCTION DERIVATIVES 5.25 REF1
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
```

```
      COMMON /BLOCK6/ CT,ST
      COMMON /BLOCK3/ H0,ALFA,KSWIT
C
      HHR = 0.0
      HHZ = 0.0
      HHP = 0.0
      THR = 0.0
      THZ = 0.0
      D = 1.0E-02
C
      IF(ABS(ALFA).LT.1.0E-06) THEN
          ALFA = 1.0E-06
      END IF
C
      FL = 0.5 * (1.-ALFA)
C
C
      DO 100 I = 1,15
C                     ARGUMENTS OF BESSEL FUNCTIONS
          ARX  = ZE(I)/R*RHO
          TARX = ZO(I)/R*RHO
C
          ARZ  = ZE(I)/R*Z
          TARZ = ZO(I)/R*Z
C
          FLZX = ALFA * ARZ
          FLZY = ALFA * TARZ
C
          EX  = 0.0
          TEX = 0.0
          IF((ARZ.LT.170.).AND.(TARZ.LT.170.)) THEN
              EX  = EXP(-ARZ)
              TEX = EXP(-TARZ)
C
              EX  = EX  * FL * AX(I)
              TEX = TEX * FL * BX(I)
          END IF
C
          FLEX = 0.0
          FLEY = 0.0
          IF ((FLZX.LT.170.).AND.(FLZY.LT.170.)) THEN
              FLEX = EXP(-FLZX)
              FLEY = EXP(-FLZY)
          END IF
C
          FLEX = FLEX * ALFA * AX(I)
          FLEY = FLEY * ALFA * BX(I)
C
          B0  = 1.0
          B1  = 0.0
          B2  = 0.0
          TB0 = 1.0
          TB1 = 0.0
C
          IF(ABS(RHO).GE.1.0E-06) THEN
C                     GET BESSEL FUNCTION
              B0  = BFJ0(ARX)
              B1  = BFJ1(ARX)
              B2  = BFJ2(B1,B0,ARX)
              TB0 = BFJ0(TARX)
              TB1 = BFJ1(TARX)
          END IF
C
          HHR = HHR + (B0-B2) * (FLEX + EX)
```

```
          HHP = HHP + (B0+B2) * (FLEX + EX)
          HHZ = HHZ +  2.*B1  * (FLEX/ALFA + EX)
C
          THR = THR + TB1 * (FLEY + TEX)
          THZ = THZ + TB0 * (FLEY/ALFA + TEX)
C
 100  CONTINUE
C               SUPERPOSITION OF 0 AND 90 DEGREE TILT CASES
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS* SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C
      HHR = +HHR -THR
      HHP = -HHP
      HHZ = -HHZ -THZ
C
      RETURN
      END
C *************************************:    VACTA
      SUBROUTINE VACTA(Z,RR,HHR,HHZ,HHP)
C
C    PURPOSE-   CALCULATE NIGHTSIDE (ZV>0) VACUUM FIELD COMPONENTS OF
C               TAIL SOLUTION FROM POTENTIAL 5.26 REF1.  SOLUTION
C               INCLUDES DIPOLE PLUS CHAPMAN-FERRARO. FIRST CALLED ON
C               SECOND ITERATION AS FIRST ITER. HAS ANALYTIC COEFFICIENT
C    INPUT-     Z,RR  Z AND RHO OF CYLINDRICAL SYSTEM
C    OUTPUT-    HHR,HHZ,HHP  FIELD COMPONENTS IN CYLINDRICAL SYSTEM
C    VARIABLES- ZE,ZO  ZEROS OF DERIVATIVES OF BESSELS J1 AND J0
C               AX,BX  COEFFICIENTS OF FOURIER-BESSEL SERIES FROM TAILCO
C                      5.24 AND 5.25 REF1
C               R, RADIUS OF TAIL
C               SNPS,CSPS  SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C               SNPH,SNPH  SIN AND COS OF PHI, A CYLINDRICAL COORDINATE
C
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT,ST
C
      HHR = 0.0
      HHZ = 0.0
      HHP = 0.0
      THR = 0.0
      THZ = 0.0
      D = 1.0E-02
C
C
      DO 100 I = 1,15
C                    ARGUMENTS FOR BESSEL FUNCTIONS
          ARX = ZE(I)/R*RR
          ARZ = ZE(I)/R*Z
          TARX = ZO(I)/R*RR
          TARZ = ZO(I)/R*Z
C
          EX  = 0.0
          TEX = 0.0
C
          IF((ARZ.LT.170.).AND.(TARZ.LT.170.)) THEN
              EX = EXP(-ARZ)
              EX = EX*AX(I)
```

```
            TEX = EXP(-TARZ)
            TEX = TEX*BX(I)
        END IF
C
        B0  = 1.0
        B1  = 0.0
        B2  = 0.0
        TB0 = 1.0
        TB1 = 0.0
C
        IF(ABS(RR).GE.1.0E-06) THEN
            B0  = BFJ0(ARX)
            B1  = BFJ1(ARX)
            B2  = BFJ2(B1,B0,ARX)
            TB0 = BFJ0(TARX)
            TB1 = BFJ1(TARX)
        END IF
C
        HHR = HHR + (B0-B2)*EX
        HHP = HHP + (B0+B2)*EX
        HHZ = HHZ + 2.0*B1*EX
C
        THR = THR + TB1*TEX
        THZ = THZ + TB0*TEX
C
 100  CONTINUE
C               SUPERIMPOSE 0 AND 90 DEGREE TILT CASES
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS* SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C
      HHR = +HHR -THR
      HHP = -HHP
      HHZ = -HHZ -THZ
C
      RETURN
      END
C ***********************************:   TAPOT
      SUBROUTINE TAPOT(NDIM)
C
C    PURPOSE-   GET SUM OF CHAPMAN-FERRARO AND DIPOLE POTENTIALS AT NDIM
C               LOCATIONS ON THE CYLINDER CAP ZV=0
C    INPUT-     NDIM, NUMBER OF GRID POINTS FOR INTEGRATION
C    OUTPUT-    W1,WO  SUM OF POTENTIALS FOR TILTS 0 AND 90 DEGREE
C                      OUTPUT IS POTENTIAL*(1/DIPOLE MOMENT0
C    VARIABLES- HS, GRID SIZE
C               R, RADIUS OF TAIL
C               X, RADIAL POSITIONS OF GRID POINTS
C
      COMMON /BLOCK8/ W1(1001),W0(1001)
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK1/ B, R, Q, PI, PSI
C
C               GENERATE GRID POINTS ON CYLINDER CAP
      X(1) = 0.0
      HS = R/FLOAT(NDIM-1)
      DO 1  I = 2, NDIM
          X(I) = FLOAT(I-1) * HS
  1   CONTINUE
C
C               GET POTENTIALS OF CHAPMAN-FERRARO TILT 0 AND 90 DEGREES
```

```
C                  EQ. 5.18-5.20 REF1
      DO 2  I = 1, NDIM
         CALL CFPOT(X(I),POT1,POT0)
         W1(I)  = POT1
         W0(I)  = POT0
  2   CONTINUE
C
C               GET POTENTIALS OF DIPOLE,TILT 0 AND 90 DEGREES 5.17 REF1
      DO 3  I = 1, NDIM
         CALL DIPOT(X(I),POT1,POT0)
         W1(I)  = W1(I)  + POT1
         W0(I)  = W0(I)  + POT0
  3   CONTINUE
C
C
      RETURN
      END
C ***********************************:    CFPOT
      SUBROUTINE CFPOT(RR,POT1,POT0)
C
C    PURPOSE-    CALCULATE CHAPMAN-FERRARO POTENTIAL FOR A POINT AT THE
C                CYLINDER LID, ZV=0. USES 5.11 WITH THETA=90 DEGREES,
C                EQUIVALENT TO 5.18-5.20 REF1. BOTH TILT CASES 0 AND 90
C                DEGREES DONE.
C    INPUT-      RR, RHO IN CYLINDRICAL, DISTANCE FROM TAIL AXIS
C    OUTPUT-     POT1, POTENTIAL FOR 0 DEGREE TILT CASE
C                POT0, POTENTIAL FOR 90 DEGREE TILT CASE
C                      OUTPUT IS ACTUALLY POTENTIAL*(-1/DIPOLE MOMENT)
C    VARIABLES- QR, DIPOLE MOMENT
C                P1,PO LEGENDRE POLYNOMIAL TERMS FOR M=1 AND 0, RESPECT.
C                G1,GO COEFFICIENTS OF LEGENDRE TERMS, M=1 AND 0 RESPECT.
C                R, RADIUS OF TAIL
C
      COMMON /BLOCK1/ B, R, Q, PI, PSI
      COMMON /CHABLO/ G1(20), G0(20)
C
      DIMENSION P0(21), P1(21)
C
C                LEGENDRE TERMS WITH THETA=90 DEGREES
      P0(2) = - 0.5
      P1(1) = + 1.0
      P1(3) = 3.*P0(2)
C
C
      DO 1  NY = 2, 10
C                EVERY OTHER TERM IS ZERO, RECURRENCE RELATION USED
C                SUBROUT. LEGEN NOT CALLED
         ZFNY = 2.*FLOAT(NY)
         FAKNY = (ZFNY-1.)/ZFNY
         P0(2*NY)    = - P0(2*NY-2) * FAKNY
         P1(2*NY+1) = + (ZFNY+1.)   * P0(2*NY)
  1   CONTINUE
C
C                TILT = 0 DEGREES CASE
      RRP = RR
      RP  = (RR*RR)/(R*R)
C
      POT1 = G1(1) * P1(1) * RR
C
      DO 2  NY = 1, 9
         ZFE = FLOAT(2*NY+1)
         RRP = RRP * RP
         POT1 = POT1 + G1(2*NY+1) * P1(2*NY+1) * RRP / ZFE
  2   CONTINUE
```

```
C
      QR = Q * (R**3)
      POT1 = -POT1/QR
C
C                 TILT = 90 DEGREE CASE
      RRP = R
      RP  = (RR*RR)/(R*R)
C
      POT0 = 0.0
C
      DO 3  NY = 1, 10
          ZFE = FLOAT(2*NY)
          RRP = RRP * RP
          POT0 = POT0 + G0(2*NY) * P0(2*NY) * RRP / ZFE
  3   CONTINUE
C
      QR = Q * (R**3)
      POT0 = -POT0/QR
C
      RETURN
      END
C ***********************************:   DIPOT
      SUBROUTINE DIPOT(RR,POT1,POT2)
C
C    PURPOSE-   CALCULATE DIPOLE POTENTIAL AT LID OF CYLINDER USING 5.17
C               REF1 FOR TILT 0 AND 90 DEGREE CASES
C    INPUT-     RR, RHO IN CYLINDRICAL, DISTANCE FROM TAIL AXIS
C               B, = RADIUS MINUS STAND-OFF DISTANCE
C    OUTPUT-    POT1,POT2 DIPOLE POTENTIAL FOR TILT 0 AND 90 DEGREES
C                         OUTPUT IS POTENTIAL*(-1/DIPOLE MOMENT)
C
      COMMON /BLOCK1/ B, R, Q, PI, PSI
C
      RRB = SQRT(RR*RR + B*B)
      RRB = RRB**3
C
      POT1 = + RR / RRB
      POT2 = +  B / RRB
C
      RETURN
      END
```

Appendix E

MSM Reference List: Correlation Studies

Compiled by R. V. Hilmer. Reference numbers correspond to the numbers on associated source chart. Note that "*author" means copy available

No.	First Author	Source
10	Akasofu, S.-I.	Magnetospheric Substorms: A Model, in Solar-Terr.Physics Part 3, ed. Dyer, 131-151, D. Reidel, 1972
20	Akasofu, S.-I.	Physics of Magnetospheric Substorms, Reidel, 1977
30	Akasofu, S.-I.	Interplanetary Energy Flux Associated with Magnetospheric Substorms, PSS, 27, 425, 1979
40	Akasofu, S.-I.	Relationships Between the AE and Dst Indices During Geomagnetic Storms, JGR, 86, 4820-4822, 1981
50	Akasofu, S,-I.	Dependence of the Geometry of the Region of Open Field Lines on the Interplanetary Magnetic Field, PSS, 29, 803, 1981
60	Akasofu, S.-I.	Energy Coupling Between the Solar Wind and the Magnetosphere, Space Sci. Rev., 28, 121, 1981
70	Akasofu, S.-I.	A High Time Resolution Study of the Solar Wind-Magnetospheric Energy Coupling Function, PSS, 30, 537, 1982
80	Akasofu, S.-I.	Dependence of the Polar Cap Geometry on the IMF,PSS,32,111-118, 1984
90	Aoki, T.	Influence of the Dipole Tilt Angle on the Development of Auroral Electrojets, J. Geomagn. Geoelect., 29, 441, 1977
100	Arnoldy, R. L.	Signature in the Interplanetary Medium for Substorms, JGR, 76, 5189, 1971
110	Aubry, M. P.	Magnetotail Changes in Relation to the Solar Wind Magnetic Field and Magnetospheric Substorms, JGR, 76, 4381, 1971
120	Baker, D. N.	High-Energy Magnetospheric Protons and Their Dependence on Geomagnetic and Interplanetary Conditions, JGR, 84, 7138, 1979
130	*Baker, D. N.	A High Time Resolution Study of Interplanetary Correlation with AE, GRL, 8, 179-182,1981.
140___	*Baker, D. N.	An ISEE-3 High Time Resolution Study of Interplanetary Parameter Correlations with Magnetospheric Activity, JGR, 88, 6230-6242, 1983
150	Ballif, J. R.	Further Evidence on the Correlation Between Transverse Fluctuations in the Interplanetary Magnetic Field And Kp, JGR, 74, 2289-2301, 1969
160	*Bargatze, L. F.	Magnetospheric Impulse Response for Many Levels of Geomagnetic Activity, JGR, 90, 6387-6394, 1985
170	Bobrov, M. S.	Kp Index Correlations with Solar-Wind Parameters During the First and Second Stages of a Recurrent Geomagnetic Storm, PSS, 21, 2139-2147, 1973
180	Bobrov, M. S.	Solar Wind Parameters Responsible for the Plasma Injection in the Ring Current, Astronomical Journal, 54, 1335,1977 ???
190	Bobrov, M. S.	Variations of the Ring Current Field in Time of Being of the Earth in the High-Speed Solar Wind, Geomagn. Aeron., 21,1048, 1981

200	Burch, J. L.	Precipitation of Low-Energy Electrons at High-Latitudes: Effect of Interplanetary Magnetic Field and Dipole Tilt Angle, JGR, 77, 6696-6707, 1972
210	Burch, J. L.	Rate of Erosion of Dayside Magnetic Flux Based on the Quantitative Study of the Dependence of Polar Cusp Latitude on Interplanetary Magnetic Field, Radio Science, 8, 955, 1973
220	Burch, J. L.	Effects of Interplanetary Magnetic Sector Structure on Auroral Zone and Polar Cap Magnetic Activity, JGR, 78, 1047,1973
230	Burch, J. L.	Observations of Interactions Between Interplanetary and Geomagnetic Fields, Rev. Geophys. Space Phys., 12, 363, 1974
233	Burch, J. L.	Effects of the Interplanetary Magnetic Field on the Auroral Oval and Plasmasphere, Space Sci. Review, 23, 449, 1979
240	Burton, R. K.	An Empirical Relationship Between Dst and the Solar Wind Electric Field and Dynamic Pressure, in Proc. of Chapman Memor. Symp. of Magneto. Motions, ed. Campbell et al., 139-143, NOAA, Boulder, Colo., 1973
250	*Burton, R. K.	An Empirical Relationship Between Interplanetary Conditions and Dst, JGR, 80, 4204-4214, 1975
260	Caan, M. N	Solar Wind and Substorm-Related Changes in the Lobes of the Geomagnetic Tail, JGR, 78, 8087-8096, 1973
270	Caan, M. N.	Substorm and Interplanetary Magnetic Field Effects on the Geomagnetic Tail Lobes, JGR, 80, 191-194, 1975
280	Caan, M. N.	Characteristics of the Association Between the Interplanetary Magnetic Field and Substorms, JGR, 82, 4837-4842, 1977
283	Caan, M. N.	Interplanetary Magnetic Field Changes and the Magnetotail, in Dynamics of the Magnetosphere, ed. Akasofu, 63-75, D. Reidel, 1979
290	Candidi, M.	Intensity Distribution of Dayside Polar Soft Electron Precipitation and the IMF, PSS, 31, 489-498, 1983
300	*Carbary, J. F.	Relations Between the Interplanetary Magnetic Field Bz, AE Index, and Cusp Latitude, JGR, 91, 1549, 1986
310	*Carbary, J. F.	Correlation of Cusp Latitude With Bz and AE(12) Using Nearly One Year's Data,JGR, 91, 10047-10054, 1986
320	*Carpenter, D. L.	Relations Between the Dawn Minimum in the Equatorial Radius of the Plasmapause and Dst, Kp, and Local K at Byrd Station, JGR, 72, 2969-2971, 1967
330	*Chappell, C. R.	A Study of the Influence of Magnetic Activity on the Location of the Plasmapause as Measured by OGO-5, JGR, 75, 50-56, 1970
340	*Chubb, T. A.	Observations of the Aurora in the Far Ultraviolet from OGO 4, JGR75, 1290-1311, 1970
350	*Clauer, C. R.	Solar Wind Control of Auroral Zone Geomagnetic Activity, GRL, 8, 915-918, 1981
360	Clauer, C. R.	Solar Wind Control of the Low-Latitude Asymmetric Magnetic Disturbance Field, JGR, 88, 2123-2130, 1983
370	Crooker, N. U.	On the High Correlation Between Long-Term Averages of Solar Wind Speed and Geomagnetic Activity, JGR, 82, 1933-1937, 1977
380	*Eather, R. H.	Dayside Aurora and Relevence to Substorm Current Systems and Dayside Merging, JGR, 84, 3339, 1979

382	*Eather, R. H.	Polar Cusp Dynamics, JGR, 90, 1569, 1985
388	Fairfield, D. N.	Average and Unusual Locations of the Earth's Magnetopause and Bow Shock, JGR, 76, 6700, 1971
390	Fairfield, D. N.	NASA GSFC-Preprint, X-690-73-366, 1973
400	*Fairfield, D. N.	Solar Wind Control of Magnetospheric Pressure (CDAW 6), JGR, 90, 1201-1204,1985
402	*Fairfield, D. N.	Polar Rain: Solar Coronal Electrons in the Earth's Magnetosphere, JGR, 90, 4055-4068, 1985
410	Feldstein, Y. I.	Dynamics of Auroral Belt and Polar Geomagnetic Disturbances, PSS, 15, 209,1967
420	*Feldstein, Y. I.	Auroras and Associated Phenomena, in Solar Terr. Physics, Part 3, ed. Dyer, 152-191, D. Reidel, 1972
427	Feldstein, Y. I.	Electric Fields and Potential Patterns in the High-Latitude Ionosphere for Different Situations in Interplanetary Space, PSS, 32, 907-923, 1984
430	*Feldstein, Y. I.	Ring Current Simulation in Connection with Interplanetary Space Conditions, PSS, 32, 975-984, 1984
440	*Feynman, J.	Prediction of Geomagnetic Activity on Time Scales of One to Ten Years, Rev. Geophys. & Space Phys., 24, 650-666, 1986
450	Foster, J. C.	Relationship of Interplanetary Parameters and Occurrence of Magnetospheric Substorms, JGR, 76, 6971, 1971
460	Freeman, J. W.	Kp Dependence of the Plasma Sheet Boundary, JGR, 79, 4315-4317, 1974
470	Fukushima, N.	Partial Ring Current Models for Worldwide Geomagnetic Disturbances, Rev. Geophys. & Space Phys., 11, 795-853, 1973
480	Garrett, H. B.	The Influence of the Variance of the Interplanetary Magnetic Field on Geomagnetic Activity, Ph. D. Thesis, Rice U., Dec. 1973
490	*Garrett, H. B.	Influence of Solar Wind Variability on Geomagnetic Activity, JGR, 79, 4603-4610, 1974
495	*Greenspan	Simultaneous Polar Cap and Magnetotail Observations of Intense Polar Rain, JGR, 91, 11123-11132, 1986
500	*Gussenhoven,M.,	DMSP/F2 Electron Observations of Equatorward Auroral Boundaries and Their Relationship to Magnetospheric Electric Fields, JGR, 86, 768, 1981
502	*Gussenhoven,M,	Systematics of the Equatorward Diffuse Auroral Boundary, JGR, 88, 5692-5708, 1983
510	Hakamada, K.	Influence of the East-West Component of the Interplanetary Magnetic Field on the Intensity of the Auroral Electrojet, PSS, 28, 29, 1979
520	*Hardy, D. A.	Response of Magnetotail Plasma at Lunar Distance to Changes in the Interplanetary Magnetic Field, Solar Wind Plasma, and Substorm Activity, JGR, 84, 1382-1390, 1979
530	*Hardy, D. A.	DMSP/F2 Electron Observations of Equatorward Auroral Boundaries and Their Relationships to the Solar Wind Velocity and the North-South Component of the Interplanetary Magnetic Field, JGR, 86, 9961-9974, 1981
540	Heppner, J. P.	Polar Cap Electric Field Distributions Related to the Interplanetary Magnetic Field Direction, JGR, 77, 4877, 1972

550	Heppner, J. P.	High Latitude Electric Fields and the Modulations Related to Interplanetary Magnetic Field Parameters, Radio Sci., 8, 933, 1973
560	*Hirshberg, J.	Interplanetary Field and Geomagnetic Variations- A Unified View, PSS, 17, 1183-1206, 1969
580	*Holzer, R. E.	An Evaluation of Three Predictors of Geomagnetic Activity, JGR, 87, 2558-2562, 1982
590	Holzworth, R.	Mathematical Representation of the Auroral Oval, GRL, 2, 377, 1975
600	*Iyemori, T.	Impulse Response of Geomagnetic Indices to Interplanetary Magnetic Field, J. Geomag. Geoelectr., 31, 1-9,1979
610	*Iyemori, T.	Prediction of Geomagnetic Activities from Solar Wind Parameters Based on the Linear Prediction Theory, in Solar Terr. Pred. Proc., 4, ed. Donnelly, US Dept. Comm., NOAA/ERL/SEL, Boulder, Colo., 1980
620	*Jones, D. F.	A Comparison of the Geomagnetic ap Index with Fluctuations in B and V: Mariner 5, JGR, 79, 286-288,1974
630	Kamide, Y.	Dependence of the Latitude of the Cleft on the Interplanetary Magnetic Field and Substorm Activity, JGR, 81, 698, 1976
640	*Kamide, Y.	Comments on No. 580 by Holzer, JGR, 88, 4953-4954, 1983
650	Kane, R. P.	Relationship Between Various Indices of Geomagnetic Activity and the Interplanetary Plasma Paramaters, J. Atmos. Terr. Phys., 34, 1941, 1972
660	Kane, R. P.	Relationship Between Interplanetary Plasma Parameters and Geomagnetic Dst, JGR, 79, 64, 1974
670	*Kaye, S. M.	The Influence of Geomagnetic Activity on the Radial Variation of the Magnetospheric Electric Field Between L=4 and 10, JGR, 86, 863-867, 1981
680	Kendall, P. C.	A Model Midnight Meridian Magnetospheric Field, Geophys. J. R. Astros. Soc., 17, 185, 1969
690	Kivelson, M. G.	The Dependence of the Polar Cusp on the North-South Component of the Interplanetary Magnetic Field, JGR, 78, 3761-3772, 1973
700	*Kivelson, M. G.	Magnetospheric Electric Fields and Their Variation with Geomagnetic Activity, Rev. Geophys. & Space Phys., 14, 189-197, 1976
710	*Kivelson, M. G.	The Physics of Plasma Injection Events, in Dynamics of the Magnetosphere, ed. Akasofu, 385-405, D. Reidel, 1979
720	Maezawa, K.	Dependence of the Magnetopause Position on the Southward Interplanetary Magnetic Field, PSS, 22, 1443, 1974
740	Maezawa, K.	Dependence of Geomagnetic Activity on Solar Wind Parameters: A Statistical Approach, Solar Terr. Environ. Res. Japan, 2, 103, 1978
750	*Maezawa, K.	Statistical Study of the Dependence of Geomagnetic Activity on Solar Wind Parameters, in Qualitative Modeling of the Magnetosphere, ed. Olson, 1979
760	*Maezawa, K.	Solar Wind Velocity Effects on the Auroral Zone Magnetic Disturbances, Sub. to Proc. Chapman Conf. SW/Magnet. Energy Coupling, April 1985

768	Makita, K.	The Shift of the Auroral Electron Precipitation Boundaries in the Dawn-Dusk Sector in Association with Geomagnetic Activity and the Interplanetary Magnetic Field, JGR, 88, 7967, 1983
770	*Makita, K.	Long-Period Polar Rain Variations and the Solar-Wind, Sub. JGR, Oct. 1985
775	Makita, K.	Temporal and Spatial Variations of the Polar Cap Dimension Inferred from the Precipitation Boundaries, JGR, 90, 2744, 1985.
780	*Mauk, B. H.	Correlation of Kp with the Substorm-Injected Plasma Boundary, JGR, 79, 3193-3196, 1974
790	McDiarmid, I.B.	Solar Particles and the Day side Limit of Closed Field Lines, JGR, 77, 1103,1972
800	*Meloni, A. A.	On the Relationships Between Interplanetary Quantities and the Global Auroral Electrojet Index, JGR, 87, 119-127, 1982
810	Meng, C.-I.	Variations of the Magnetopause Position with Substorm Activity, JGR, 75, 3252, 1970
820	*Meng, C.-I.	Cross-Correlation Analysis of the AE Index and the Interplantary Magnetic Field Bz Component, JGR,78,617-629,1973
830	Meng, C.-I.	Polar Cap Variations and the Interplanetary Magnetic Field,in Dynamics of the Magnetosphere, ed. Akasofu, 23-46, D. Reidel, 1979
840	Meng, C.-I.	Latitudinal Variations of the Polar Cusp During a Geomagnetic Storm, GRL, 9, 60, 1982
850	Meng, C.-I.	Case Studies of the Storm Time Variation of the Polar Cusp, JGR, 88, 137, 1983
860	Mishin, V. M.	Short-term Forecasting of Geomagnetic Storms Associated with High-Speed Solar Wind Streams, in Solar-Terr. Predictions Proceedings, 4, ed, Donnelly, A37-A45, US Depart. of Commerce, Boulder, Colo., 1980
870	Mozer, F. S.	The Auroral Zone Electric Field, JGR, 79, 1001,1974
880	Murayama, T.	Effects of Solar Wind Parameters on the Development of Magnetospheric Substorms, PSS, 23, 75, 1975
890	*Murayama, T.	Empirical Formula to Relate the Auroral Electrojet Intensity with Interplanetary Parameters, PSS, 28, 803-813, 1980
900	*Murayama, T.	Coupling Function Between Solar Wind Parameters and Geomagnetic Indices, Rev. Geophys. & Space Phys., 20, 623-629,1982
910	Murayama, T.	Comparison of the Observed Values of the Dst Index with the Predicted Values by the Coupling Functions BsV, BsVV, epsilon, pBsV, Sol.-Terr. Environ. Res. Japan, 6, 39, 1982
920	*Murayama, T.	Coupling Function Between the Solar Wind and the Dst Index, Sub. to Proc. Chapman Conf. SW/Magnet. Energy Coupling, April 1985
930	Parks, G. K.	Correlation of Interplanetary-Space Bz Field Fluctuations and Trapped Particle Redistributions, JGR, 77, 266, 1972
940	Patel, V. L.	Geomagnetic Activity and Size of Magnetospheric Cavity, JGR, 71, 1940, 1966

950	Perreault, P.	A Study of Geomagnetic Substorms, Geophys. J. R. Astro. Soc., 54, 547, 1978
960	Pike, C. P.	Observed Correlations Between Interplanetary Field Variations and the Dynamics of the Auroral Oval and the High-Latitude Ionosphere, JGR, 79, 5129, 1974
970	Reiff, P. H.	Dependence of Polar Cap Potential Drop on Interplanetary Paramaeters, JGR, 86, 7639, 1981
975	*Riehl, K. B.	Average Characteristics of the Polar Rain and Their Relationship to the Solar Wind and Interplanetary Magnetric Field, JGR, 91, 1557-1571, 1986
980	*Rosenberg, R. L.	A Practical Method of Predicting Geomagnetic Storms by Using Precursory Enhancements of Solar Wind Density, JGR, 87, 2563-2568, 1982
990	Rostoker, G.	Relationship Between Changes in the IMF and Variations in the Magnetic Field at the Earth's Surface, JGR, 72, 5853-5863, 1967
1000	Rostoker, G.	Response Time of the Magnetosphere to the Interplanetary Electric Field, Canadian J. Phys., 50, 544, 1972
1002	Rostoker, G.	Variation of Auroral Electrojet Spatial Location as a Function of the Level of Magnetospheric Activity(B), JGR, 91, 1716, 1986.
1010	*Rothwell, P. L.	An Empirical Relationship Between Kp and Tail Lobe Magnetic Field, JGR, 91, 9012-9016, 1986
1020*	Rudneva, N. M.	Distance to the Subsolar Point of the Magnetosphere Boundary as a Function of Magnetic Activity, Geomagn. Aeron., Engl. Tran., 10(5), 635-638, 1970
1030	Rudneva, N. M.	Position of the Boundary of the Magnetosphere During 1963-1968, Geomagn. Aeron., 13, 96, 1973
1040	Russell, C. T.	The Solar Wind and Magnetospheric Dynamics, in Correlated Interplanetary and Magnetospheric Observations, ed. Page, 3-47, D. Reidel, 1974
1050	Russell, C. T.	The Control of the Magnetopause by the Interplanetary Magnetic Field, in Dynamics of the Magnetosphere, ed. Akasofu, 3-21, D. Reidel, 1979
1060	Rycroft, M. J.	The Magnetospheric Plasmapause and the Electron Density Trough at the Alouette 1 Orbit, PSS, 18, 65, 1970
1070	Sandholt, P. E.	Effects of Interplanetary Magnetospheric Substorm Variation on the Dayside Aurora, PSS, 31, 1345, 1983
1080	*Sandholt, P. E.	Large- and Small-Scale Dynamics of the Polar Cusp, JGR, 90, 4407 1985
1090	Schatten, K. H.	Response of the Geomagnetic Activity Index Kp to the Interplanetary Magnetic Field, JGR, 72, 5185-5191, 1967
1100	Schatten, K. H.	Large-Scale Properties of the Interplanetarty Magnetic Field, Solar Wind, NASA Spec. Publ. SP-308, 65, 1972
1110	Sergeev, V. A.	Quantitative Dependence of the Polar Cap Electric Field on the IMF Bz-Component and Solar Wind Velocity, PSS, 29, 205, 1980
1120	*Siscoe, G. L.	A Quasi-Self-Consistent Axially Symmetric Model for the Growth of a Ring Current Through Earthward Motion from a Pre-Storm Configuration, PSS, 27, 285-295, 1979

1130	Siscoe, G. L.	A Dst Contribution to the Equatorward Shift of the Aurora, PSS, 27, 997-1000, 1979
1140	*Siscoe, G.	A Theoretical Relation Between Dst and the Solar Wind Merging Electric Field, GRL, 1, 17-19,1974
1150	Siscoe, G.	On the Partial Ring Current Contribution to Dst, JGR, 79, 1110, 1974
1170	*Snyder, C. W.	The Solar Wind Velocity and Its Correlation with Cosmic-Ray Variations and with Solar And Geomagnetic Activity, JGR, 68, 6361, 1963
1175	*Sojka, J. J.	An Interplanetary Magnetic Field Dependent Model of the Ionospheric Convection Electric Field, JGR, 91, 11281-11290, 1986
1177	Stiles, G. S.	Ionosonde Observations of the Northern Magnetospheric Cleft During December 1974 and January 1975, JGR, 82, 67, 1977
1180	*Su, S.-Y.	Magnetic Field Depression at the Earth's Surface Calculated From the Relationship Between the Size of the Magnetosphere and the Dst Values, JGR, 80, 195-199,1975
1190	*Sugiura, M.	Quiet Time Magnetospheric Field Depression at 2.3-3.6 Re, JGR, 78, 3182-3185, 1973
1200	Svalgaard, L.	The Relation Between the Azimuthal Component of the Interplanetary Magnetic Field and the Geomagnetic Field in the Polar Cap, in Correlated Interplanetary and Magnetospheric Observations, ed. Page, 61-84, D. Reidel, 1974
1210	Svalgaard, L.	Geomagnetic Activity: Dependence on Solar Wind Parameters, in Coronal Holes and High-Speed Wind Streams, ed. Zirker, Colo. Assoc. U. Press, Boulder, Colo., 1977
1220	Taylor, H. A.	Contraction of the Plasmasphere During Geomagnetically Disturbed Periods, JGR, 73, 961, 1968
1230	Troshichev, O.	Interplanetary Magnetic Field and Polar Cap Magnetic Distributions: Using the Data for Predictions of Auroral Electrojet Activity, in Solar Terr. Pred. Proc., ed. Donnelly, US Dept. Comm., NOAA/ERL/SEL, Boulder, Colo., 1980
1240	*Troshichev, O.	The Relationship Between Interplanetary Quantities and Magnetic Activity in the Southern Polar Cap, PSS, 33, 415-419, 1985
1250	Tsurutani, B.	Interplanetary Magnetic Field Variations and Substorm Activity, JGR, 77, 2964, 1972
1260	Wilcox, J. M.	Influence of Interplanetary Magnetic Field and Plasma on Geomagnetic Activity During Quiet Sun Conditions, JGR, 72, 19-26, 1967
1290	Verzariu, P.	Geomagnetic Field Variations Caused by Changes in the Quiet-Time Solar Wind Pressure, PSS, 20, 1909-1914, 1972
1300	Yashuhara, F.	Equatorward Shifts of Cleft During Magnetospheric Substorms as Observed By ISIS I, JGR, 78, 7286, 1973
1310	Yoshizawa, K.	Influences of IMF-Sector Polarity on the N-S Asymmetry of Geomagnetic Activity and Its Relationship with Solar Wind Parameters, PSS, 32, 605-610, 1984

		Kp	Dst	AE	ap	Bx	By	Bz(Bs)	Bz*Bz	
1	Stand-Off	390 1020 1030	420 1020 1180	8 [illegible]	940			388		
2	Cusp Latitude	630 790	630 840 850	300 310 630 1300				200 220 300 310 630 690 850 1070 1080 1177 1300		
3	Plasma Sheet	460 710 780								
4	Plasmapause	320 330 700 1060 1220	320							
5	PC Latitudes	340 380 410 420 500 1175	10 1120 1130	380 382 775		80 830	80 830	80 380 382 530 590 830	530	
6	PC Potential	530 540 670 700 970 1175		970 1110		427	427	427 970 1110 1175		427
7	PC Precipitation	290 495 502		768 775		975	290 975	290 495 768		
8	Kp	xxxxxxxxxxxx xxxxxxxxxxxx					880	170 560		170 1090
9	Dst		xxxxxxxxxxx xxxxxxxxxxx					250 600 610 660 900 920 1040		
10	AE		40	xxxxxxxx xxxxxxxx		800 220	800 880	100 130 140 450 490 600 800 820		490
11	AL					220	510 890	510 600 610 900		740
12	AU					220		600 610		740
13	Ap	560			xxxxx xxx			370 490		490
14	Am									740
15	N(S W Density)	980	980					980		980
16	Tail Lobe Ener. D.	1010						260 270 283		520
17	U(magn. ener. dis.)		60					140		
18	Delta B(magn)		1190							

	B	V	VB	VBz	V(B*B)	(V*V)Bz	V(Bz*Bz)	N	N(V*V)	Epsilon	B
1											
2											
3											
4											
5				530			530				960
6	427 540 970	970 1110		970						970	
7		495 402						402 495 770			
8	170 560 1090 1260	170 1090 1170 1260		530		880		170 1260			150
9		250 610 660 900 920 1040	240 1140	910		900 910		250 610 660 920 1040	240 1290	910	
10	490 800	130 490 800		130 140 160 490 800		140 880		490 800	490	30 130 140 160 800	490
11	740 750 760	610 740 750 760 900		160 350 360 580 640	350	350 580 640 890 900	580 640	610 750 760 890		160 900	740 890
12	740 750	610 740 750		350	350	350		610 750			740
13	490 1090	370 490		490				490	490		490
14	740 750	740 750						750			740
15	980	980						xxxxxxxxx xxxxxxxxx			980
16	520			110					260		
17				140		140				70 140	
18											

	B Var.	Tilt	SW(T)	SW(deg)	PC Lat	SORT
1					1020	a
2		200				b
3						d
4						e
5	960	420				f
6						g
7			975			h
8	150 560		170			i
9						j
10	490			800		k
11	740 750 890	90 890	750	740 750		l
12	740 750	90	750	740 750		m
13	490 620					n
14	740 750		750	740 750		o
15	980					q
16						r
17						s
18						u